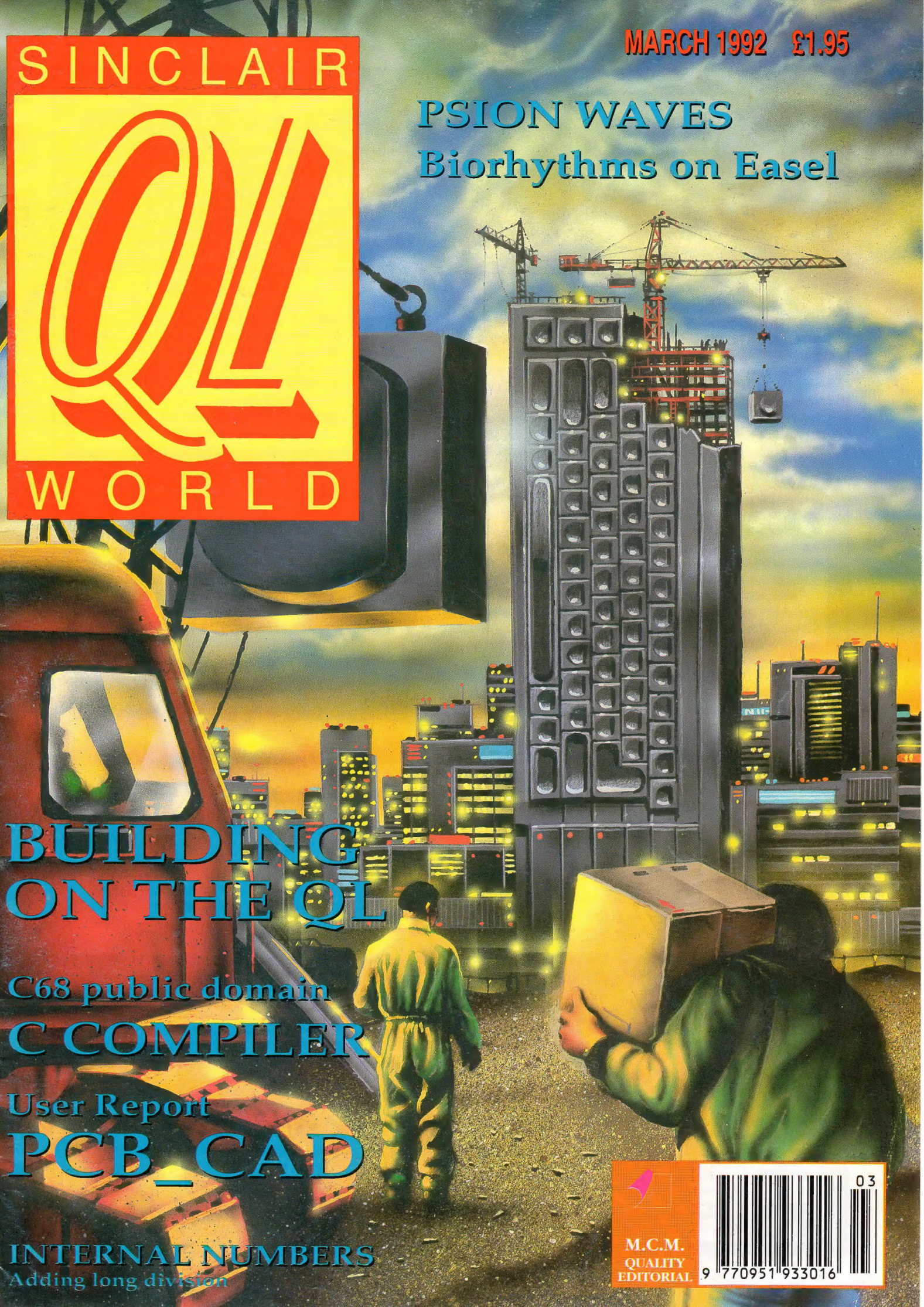

SINCLAIR
QL
WORLD
MARCH 1992 £1.95
PSION WAVES
Biorhythms on Easel
BUILDING ON THE QL
C68 public domain
C COMPILER
User Report
PCB_CAD
INTERNAL NUMBERS
Adding long division
M.C.M.
QUALITY
EDITORIAL
9 770951 933016 03

DJC

Dilwyn Jones Computing

41 Bro Emrys, Tal-y-Bont,
Bangor, Gwynedd LL57 3YT
Tel: Bangor (0248) 354023

SOFTWARE

ADDRESS LABELLER(flp, mdv, 384k, R)	£15.00
BASIC REPORTER (flp, 128k)	£10.00
BASIC REPORTER (mdv, 128k)	£12.00
BIBLE TEXT DISKS (flp, Quill format)	£20.00
BIBLE TEXT DISKS (flp, Editor format)	£20.00
COCKTAILS WAITER (flp, 256k)	£10.00
COCKTAILS WAITER (mdv, 256k)	£12.00
Cocktails waiter recipes sets:	
MIX 2 (flp, mdv)	£5.00
MIX 3 (flp, mdv)	£5.00
DATA DESIGN, by PROGS V2.03 (flp, 512k)	£50.00
DISCOVER (fllp, 256k, dual drives)	£20.00
MULTI DISCOVER (flp, 256k, dual drives)	£30.00
TEXTIDY (flp, 256k)	£15.00
FLASHBACK (flp, mdv, 256k)	£25.00
FLASHBACK SPECIAL EDITION (flp, 256k)	£40.00
Upgrade FLASHBACK to FLASHBACK SE (return master copy)	£20.00
HOME BUDGET (flp, 128k)	£20.00
HOME BUDGET (mdv, 128k)	£20.00
FILEMASTER (flp, mdv, 512k, R)	£12.00
FILES 2 (flp, mdv, 128k)	£12.00
IMAGE PROCESSOR V2 (flp, mdv, 256k)	£15.00
PAGE DESIGNER 2 PLUS (flp, 512k)	£40.00
Upgrade PD2, return master copy	£20.00
PD2 CLIPART, Screen Format (flp, 128k)	£10.00
PICTUREMASTER (flp, 256k)	£15.00
PICTUREMASTER PLUS (flp, 256k)	£20.00
Upgrade PICTUREMASTER to PLUS version (quote serial number)	£5.00
QL GENEALOGIST 2ND EDITION (flp, 384k)	£30.00
QL GENEALOGIST 1ST EDITION (flp, 384k)	£19.50
Upgrade QLG1 to QLG2 (return master)	£12.00
BUDGET QL GENEALOGIST (flp, mdv, 128k)	£12.00
Upgrade Budget QLG to QLG 2nd Edition (return master)	£24.00
QLIBERATOR COMPILER (flp, 256k)	£50.00
BUDGET QLIBERATOR (flp, mdv, 128k)	£25.00
QLOAD/QREF (flp, mdv, 128k)	£15.00
QUESTION MASTER (flp, 128k)	£10.00
QUESTION MASTER (mdv, 128k)	£12.00
QUIZ SETS for Question Master. Each	£5.00
1. General Knowledge 2. Classical Music 3. QLestions (Questions about the QL)	
QUICK POSTERS (flp, 128k)	£10.00
QUICK POSTERS (mdv, 128k)	£12.00
REMIND-ME (flp, mdv, 128k)	£12.00
ROB ROY BARGAIN PACK (flp, 128k)	£10.00
ROB ROY BARGAIN PACK (mdv, 128k)	£14.00
SCREEN SNATCHER (flp, mdv, 128k)	£12.00
SCREEN ECONOMISER (flp, 128k)	£10.00
SCREEN ECONOMISER (mdv, 128k)	£12.00
SLOWGOLD 2 (flp, mdv, 128k)	£5.00
SPELLBOUND (flp, mdv, 256k)	£30.00
SPELLBOUND SPECIAL EDITION (flp, 512k)	£50.00
Upgrade Spellbound to SE (return master)	£30.00
SUPER DISK LABELLER (flp, 256k)	£10.00
SUPER DISK INDEXER (flp, mdv, 256k)	£12.00
TASKMASTER (flp, mdv, 384k)	£25.00
Upgrade TASKMASTER for Hard Disks (return master copy)	£10.00
THE CAT (flp, 128k)	£5.00
THE CAT (mdv, 128k)	£5.00
THE CLIPART, by PROGS (flp, mdv, 128k)	£12.00
THE PAINTER V4.03, by PROGS (flp, 512k)	£25.00
THE GOPHER (flp, mdv, 128k)	£12.00
TRANS24 (flp, 128k)	£10.00
TRANS24 (mdv, 128k)	£12.00
VISION MIXER (flp, 384k)	£10.00
VISION MIXER (mdv, 384k)	£12.00
VISION MIXER PLUS (flp, 256k)	£22.50
Upgrade VISION MIXER to VISION MIXER PLUS (quote serial number)	£14.50
WINBACK (flp, 256k)	£25.00

NB 'flp' = disk, 'mdv' = microdrive, 'R' = RAMdisk reqd. 128k/512k etc. = minimum memory reqd.

LABELS

MICRODRIVE LABELS roll of 100	£2.00
DISK LABELS roll of 100	£2.00
DISK LABELS roll of 100 on printer pin feed backing	£2.50
ADDRESS LABELS printer roll of 100 3.5" wide, 1.5" high	£2.00

Please add 50p postage per roll of labels (max. postage £2.50) if only orderning labels.

FLOPPY DISKS, ETC.

3.5" unbranded DSDD disks	£0.45
20 or more disks	£0.40
Pack of 10 SONY branded 3.5"	£9.00
5.25" DSDD unbranded disks	£0.35
Microdrive cartridges (new)	£2.50

N.B. PLEASE ADD POSTAGE SPECIFIED BELOW. NO NEED TO ADD POSTAGE ON MICRODRIVE CARTRIDGES IN UK.

DISK STORAGE BOXES

Posso boxes, drawer style, holds 150 3.5" disks, stackable, storage system	£17.00
Box to hold 80 3.5" disks, with lockable, clear lid	£7.00
Box to hold 40 3.5" disks, with lockable, clear lid	£5.00
Slimpak case, small plastic box, holds 10 3.5" disks	£1.20
Set of 20 headed disk box subject dividers, "dry wipe" laminated, assorted headings one side, blank other side	£3.00

POSTAGE RATES – SEE BELOW

CATALOGUE

PLEASE ASK FOR A COPY OF OUR QL SOFTWARE CATALOGUE (SENT FREE ON REQUEST WITH ORDERS). WE NOW SELL SO MANY ITEMS THAT WE CANNOT PROVIDE MUCH DETAIL IN THE ADVERT, THE CATALOGUE CONTAINS MORE DETAILS OF ALL OUR PROGRAMS

SOFTWARE DISCOUNTS

BUY 2 PROGRAMS TOGETHER, CLAIM 5% OFF

BUY 3 OR MORE PROGRAMS, CLAIM 10% OFF

STANDS

Monitor stand, tilt & swivel, holds up to 14" screen monitors	£15.50
Printer stand, plastic, fits most 80 column printers	£7.50

POSTAGE RATES – SEE BELOW

SOFTWARE POSTAGE: Software sent post-free to UK addresses. Abroad add £1.00 per program for postage and packing. SUPPLIES POSTAGE: Add £2.50 per order to UK addresses (except labels, see above). Abroad, add 10% of order value (minimum £3.50) for airmail delivery where possible.
PAYMENT:We can accept payment by cash, cheque (in Pounds Sterling drawn on UK branch of bank or building society), Postal Order, International Postal/ Money order, Eurocheque, Payments in UK currency (Pounds Sterling), please, in cases of difficulty, contact us first to arrange payment method.
We can accept payment by credit card (VISA, ACCESS, MASTERCARD, EUROCARD). Please state your card type, number, expiry date, address (goods must be sent to cardholder's address) and remember to sign your order. Goods remain property of DJC until paid for in full. Our telephone number has an answering machine for when we are unable to answer in person – don't be shy!

VISA · Access · EUROCARD · Master Card

SINCLAIR QL WORLD

Editor
Helen Armstrong

Production Controller
Jayne Penfold

Designer
Jeff Gurney

Advertising Manager
Jason Newman

Magazine Services
Yvonne Taylor

Advertising Production
Michelle Evans

Group Advertising Manager
Lynda Elliott

Group Editor
John Taylor

Publishing Director
Wendy Palmer

Group Deputy Managing Director
Ray Lewis

Group Managing Director
Peter Welham

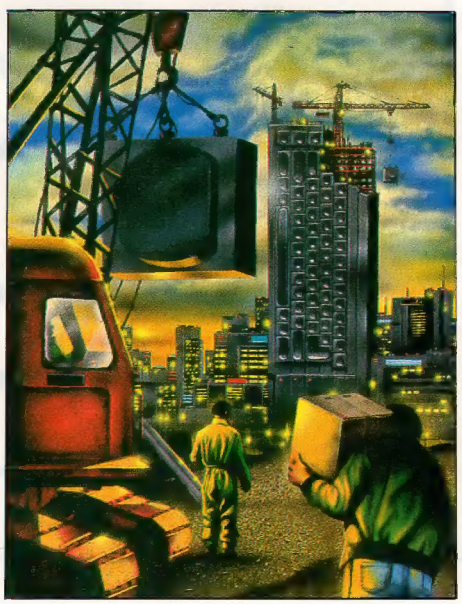
Sinclair QL World
Panini House
116-120 Goswell Road
London EC1V 7QD
Telephone 071-490 7161
ISSN 026806X

Unfortunately, we are no longer able to answer enquiries made by telephone. If you have any comments or difficulties, please write the The Editor, Open Channel, Trouble Shooter, or Psion Solutions. We will do our best to deal with your problem in the magazine, though we cannot guarantee individual replies.
Back issues are available from the publisher price £2 U.K., £2.75 Europe. Overseas rates on request.
Published by Maxwell Consumer Magazines, A Division of MCPC Ltd., Sinclair QL World is distributed by IPC Market force, King's Reach Tower, Stamford Street, London SE1 9LS.
Subscription information from: MCM Subscription Dept, Lazahold Ltd., PO Box 10, Roper St, Pallion Ind. Est., Sunderland SR4 4SN. Tel: 091 510 2290
UK: £21.00, Europe: £32.00,
Rest of the World: £38.00.
Please include your subscriber number with any queries.

Typesetting by Ford Graphics, 8-10 Whitsbury Road, Fordingbridge, Hampshire. SP6 1BR. Tel: (0425) 655657
Printed and bound by BPCC, Colchester. Covers printed by Spottiswoode Ballantyne, Colchester.
Sinclair QL World is published on the third Thursday preceding cover date.

CONTENTS

MARCH 1992



NEXT MONTH

DIY Toolkit presents array parameters and fast floating-point operations; we have that working comparison of three word-processors, which missed the bus this month, and we hope to start some 'notebook' pages of short ideas to keep relations with your QL on an even keel.

TROUBLE

A PROBLE

Bryan Davis reconsiders Winback, disturbed ribbon cables and mixed disk densities.

Harking back to comments made about the hard disk backup program *WinBack,* it has been pointed out to me that this program (and its associate, *The Gopher*) do not Open files in a potentially-dangerous fashion, as *Archive* does. The latter program actually makes a change to a file when it Opens it, whereas neither of the other two programs does anything to the contents of an Opened file. As noted previously, it seems likely that an aborted backup attempt, using Winback, was the result of a bad area on the hard disk, rather than a corrupted file as such; had the problem been simple file corruption, it would have been possible to re-copy the file when the backup halted, then re-commence the backup where it halted. If the process is restarted, WinBack passes over all the files which have already been backed up and starts copying at the file which caused the stoppage. Make sure the same backup disk is in the floppy drive.

Preliminary V2

A preliminary version 2 of Winback has been tested, but it did not have the file-splitting capability which will be in the released version. The intention is for the program to split single files between two disks, if the end of one disk is reached during backing up of a file. For the moment, file compression is not planned for the good reason that such a feature is fraught with potential danger for the user's files. In the long run, compression may be added, as it is otherwise a very desirable feature. At least one file-compression routine is already available for the QL, supplied with the C68 compiler system. Those who are fortunate enough to have ED drives will not be too concerned about compression, since they won't need many disks to backup a hard drive.

A 'dead' QL was brought to me for investigation recently. When it was switched on, the mono screen glowed with an unsteady light green and there was no sign of the F1/F2 start-up display. There was an obvious choice of two areas to look for the fault – either the Qimi mouse interface was playing up, or the ZX8301 chip which handles the display interface was kaput. On the basis that one of the owner's Qimis had caused trouble before, and this one might have to be 'got at' to prevent conflict with a Gold Card anyway, it was removed, but the display was still useless. Changing the ZX8301 (the chip to the right of the large cpu chip) fixed the display, and the F1/F2 screen appeared, but keying F1 had no effect. You can make a fair bet that this will happen whenever the QL is opened up, the cause being disturbance of the ribbon connectors to the keyboard. They don't make the best connections, and kink easily when re-inserted. Kinking may break a lead. If a bit at the connector end has been bent over, cut it off cleanly with scissors and re-insert it. As an extra precaution, spray the terminals inside the connectors with *contact spray* – that is, a cleaner designed specially for electrical contacts, **not** any old spray.

When you buy a replacement for a ZX8301 chip, it may come labelled CLA2310, which is alright; however, sometimes a particular ZX/CLA chip is unhappy working with the 68008 cpu chip – it might be necessary to do some chip matching. As to what causes 8301 chips to blow, there are a few theories, the most common being that the unbuffered connection between the chip and the display unit allows excess voltages from the latter to overload the chip. In the early days of the QL, we were warned by peripheral equipment manufacturers to switch on heavy-current units such as displays and printers *before* the QL was switched on, and to switch off the QL before the other units. This procedure ensures that the QL is 'dead' whenever excess voltage is fed to interconnecting wires. My own procedure for nearly seven years has been to have all units connected to the same multi-way extension socket and switch them all on and off together. That has not given any problems, **but** I must point out that my system has a variety of suppression devices fitted and there can, anyway, be no assurance that simultaneous switching will be satisfactory with any combination of units. The speed with which some units 'come alive' could be much greater than for others. One thing you should note, if failure of the 8301 has occurred several times on your QL(s), is whether or not it happened when the system was switched on, rather than while it was working.

Voltage surge

If you have seen the blank screen only at switch on time, there is certainly good reason to suspect a voltage surge from a peripheral unit as the culprit; don't forget that anything plugged into the 64-way expansion port is a peripheral unit – the display may not be the problem area.

72-point
72-point

SHOOTER

M S O L V E D

Borrowing is ok, for a while, but one can't expect the owner of a Gold Card to loan it out indefinitely; fortunately, I've finally got my own GC and can investigate what it does for the QL, on a day-to-day basis. Previous attempts to get a GC and my one-third height Mitsubishi 3.5in drives to co-operate had been unsuccessful, despite a variety of settings for the FLP_STEP 5 parameter. Other users had advised changing FLP_START instead, so that's what I did with the new GC. It didn't work. Back to FLP_STEP, just in case . . . It was a good job the check was made, because the current GC – identified as 'Gold Card 2' underneath and with the pcb coloured red – seems to get on fine with these drives when FLP_STEP is set (the default is 3); the FLP_START parameter was left unchanged, at the default 30.

Combined disks

Mention was made in the February issue of combining ED (extra-high density) or HD (high-density) and DD (double-density) 3.5 in floppy disk drives with a Gold Card interface. The adapter which has to be fitted to the GC to enable this has two drive connectors, one pointing upwards, the other outwards (in the 'conventional' position). To avoid the necessity for changing over drive designation jumpers (the ones that identify drives as flp1_, flp2_ etc), the straight-out connector is understood by the QL to connect to flp3_ and flp4_. That is, although the DD drives were previously always flp1_ and flp2_ when connected as the only two drives on the system (with either Trump Card or Gold Card), they have now become flp3_ and flp4_, without having to be fiddled with. The ED drives are plugged into the upwards pointing connector, and they have become flp1_ and flp2_; their internal jumpers also are set to flp1_ and flp2_. As mentioned above, the FLP_STEP parameter has had to be reset, by a line in the system boot program, but that was solely to cure a disagreement between GC and DD drives. Otherwise, no fiddling was necessary, and all four drives are working fine. One (branded) ED disk failed to format first time, but was alright on the next try. At £30 for 10, these disks are still rather pricey, but the value in terms of bytes per penny is good.

There are some warnings given in the instructions with the GC and ED drives, and they bear repeating here. If ED drives are connected to a Trump Card, they can be used only as DD drives (720 KB) and the TC has to be fitted with the version 1.34 or later rom. When using either HD or ED disks in DD drives, cover over (with opaque, not transparent, tape) the square cutout opposite the write-protect cutout. DD, HD and ED disks have a write-protect cutout and slide in the same location, but HD and ED disks have an additional cutout on the opposite side (but in slightly different positions). By covering up this cutout, you should prevent DD drives from failing to format the HD or ED disks properly. Don't be fooled by an apparently-correct format operation without the cutout taped over; my own quick check did show both HD and ED disks to format properly in a DD drive, and it was possible to copy files to them from hard disk, then delete them, but neither disk was recognised when put into an ED drive. That is, you can do the formats but don't expect compatibility between drive types unless the cutout is taped over. There may be some incompatibility even when the cutouts are taped over; while various WCOPY operations between drives succeeded on my system, others failed with the 'bad or changed medium' message being displayed, and there was no reason to suspect the disks being used. It may be advisable to match disks to drives – that is, DD disks in DD drives, and so on.

Rom version

An ED disk formatted in a HD drive should give 1.44MB; presumably, the same advice – cover the cutout – applies here, too. For ED drives to work properly with the GC, the rom in the latter must be version 2.22 or later. Remember that, although the GC can provide sub-directories on hard and floppy disks, the TC (and other interfaces) cannot deal with sub-directories on **floppy** disks, so don't switch disks which have sub-directories on them to systems which don't have a Gold Card.

Density needed

My initial reason for wanting ED drives was the prospect of being able to load all system files from one disk. A single DD disk had long since proved too small to hold more than the bare necessities, but a disk with more than four times the capacity would be ample (for a while). A more pressing reason presented itself shortly before the ED drives arrived; something had gone wrong with the 20 MB MS-DOS 'partition' of the hard disk, *Conqueror* couldn't start, and a backup of the hard disk was called for as a precaution before the partition was reformatted. The space taken by Qdos files had risen to over 12 MB and it would have taken near to twenty DD disks to hold the backup copies; the actual number required varies with the backup program being used, and the position of files on the hard disk. If the backup program attempts to fill each disk, the number is kept down but, if it merely copies files in sequence, a big file which doesn't quite fit the current disk will cause the remaining space to be unused.

Hardback

Back in the days of PDQL, a copy of Chas Dillon's *HardBack* was sent for review, but had been put on one side for obvious reasons; now was a good time to try it. The 12.324 MB of files fitted comfortably onto four ED disks, leaving 456 KB of space unused. On the first three disks, only 2 KB remained unused, showing that the program had done an excellent job of filling the disks. As the data transfer rate for the ED drives is in the region of 1 megabyte per second, compared to about a quarter of that with the DD drives, the job was completed in well under a half hour. Subsequent attempts to make back-ups of only those files which have changed since the full backup have not been successful; resolution of that problem will have to wait upon some spare time to read the instructions again.

Many micro users will be conscious of some slowness in their computer's handling of certain jobs. The QL is perhaps a bit unusual in that it was created with a slow word processor program, *Quill*. Generally, handling of text should be relatively fast, whereas graphics can be expected to be dealt with at a snail's pace. It was unfortunate that Psion had to produce Quill at a time when they did not have anything like a 'final' QL available to test it on. It seems likely that – given time and money – they would have completely rewritten the program. That is history, but users have now got the option to replace Quill with *Perfection*, and get both greater

speed and more facilities, and/or install a Gold Card and have all programs run faster.

It really is a pleasure to see things happening much faster on the screen when a GC is fitted. In my own case, arrival of the GC coincided with a very frustrating period trying to cope with the PC wonder-program *Windows*, and the contrast was marked. Using a WP program under Windows, on a reasonably-fast PC, there is frequently enough time available to have a cup of coffee – maybe a light meal! – while one waits for some basic operations to be completed, whereas there are no significant pauses using Perfection with the GC on the QL. The latter combination looks, and is, fast; the sluggishness of two WP programs tried with Windows was such as to make it out of the question to use them on a daily basis. On top of this, there has been a period of eight weeks waiting for the world's #1 software company (you guess the name) to explain how to make one of the WP programs work at its proper (slow) speed. We generally get better service than this on the QL.

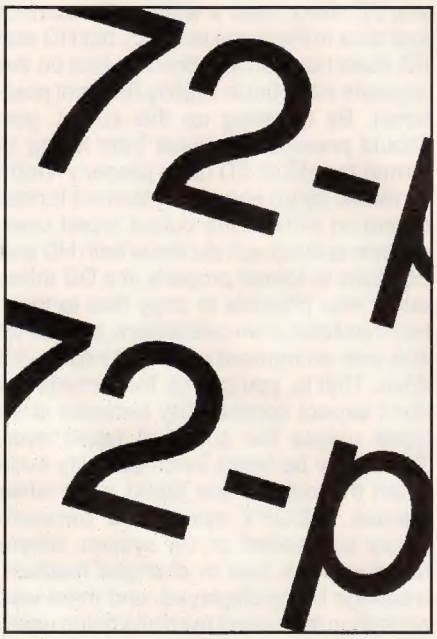
Catching up

QL software and firmware has yet to catch up with the hard disk, high-density drives and sub-directories. Since adding these to my system, there has been a lot of aggravation, stemming from commands which don't work properly and programs which are reluctant to be configured to work from hard disk. The wild-card commands appear to work as and when they feel like it; WDEL *seems* to work, only for deleted files to reappear later, sometimes after several 'deletions', and WSTAT says 'not found' to sub-directory files. Whether the problems noted relate to the commands themselves, to the interface or drives, to the use of a device driver to make sub-directories more usable, or to some other factor, I know not. Reconfiguring *text87* (version 3.00) and *Professional Publisher* has been a pain. Conqueror itself was ok, but the partition it was using on the hard disk became unusable. Can we hope that programmers will start to make provision for hardware that has been available on QL systems since 1990-1991?

In case you are confused about the number of floppy disk drives you can connect to your Trump Card or Gold Card, note that there are **two** adapters sold by Miracle for increasing the number of drives supported. Only two drives could be connected to earlier TCs, and the standard adapter for these cards increases the number to four. Later TCs supported four drives, so don't need an adapter. The GC supports three drives, and the adapter for it raises this to four; this same adapter can be used on TCs, for the 2-4 conversion. There is only slight difference between the adapters, but the original one is not suitable for the GC; specify your interface type when ordering.

Recognition needed

Qdos can support eight drives, but the interface specification for disk drives does not normally permit this number. It is interesting to read that even the scsi interface, which should support many chained devices (of varying types), does not actually make it straightforward – in practice – to connect additional disk drives beyond the usual two or three. Miracle report that one



of the Select lines on the ED drives is used for selecting density, thereby reducing the possibilities for drive selection. At the time of writing, my one QL is happily recognising two ED and two DD drives when a Gold Card is the interface, but another QL recognises only one of each type of drive when a Trump Card is the interface. Bear in mind that the TC does not support HD or ED drives, except when they are used with DD-formatted disks.

Height of confusion

When referring to character heights in picas (one-sixth of an inch, a normal measure in the print trade), there is a fair chance of the inexperienced user getting confused, if he/she actually measures the height of printed characters. The normal height figures given, for example, Helvetica 72-point, refer to the total space taken by the characters, including space top and bottom, not to the height of the characters themselves. A 'point' is one-twelfth on a pica. Refer to the illustration on the previous page and the point should be clearer; there has to be space above and below, to avoid characters on successive lines touching each other, and a **cell** height is specified to allow for this. The character height is appreciably less than the cell height. In the case of text 87's Roman 21-point fount, the characters are roughly 15 points high. Additionally, printers do not necessarily use the same data for all founts of the same name. For example, there are a large number of variations of the Times fount, and you cannot be sure of the character point size for a particular printer set, without measuring the printed output. Equally, don't expect characters of the same nominal point size, but of different founts, to be the same height. They vary enormously! The typical Times and Helvetica sizes used for correspondence fortunately differ only slightly in height; to get the same character height you have to use about 10.5 points for Times against 10 points for Helvetica.

The super-large dictionary for Digital Precision's *Spellchecker* will be supplied on two DD disks, rather than on one HD or ED disk. To enable this, files have been created for each of the alphabetic letters and a reconstitution routine has been provided to stitch them all together again as one dictionary. This routine actually has two options – make one complete dictionary, or two separate ones, more or less equal in size. It does the job quite quickly. Incidentally, don't think development has stopped on *Perfection*; my current version is 3.07. There have not been major functional changes in the last few versions, so users of earlier versions need not get worried about missing something.

Readers' letters

If **D. Shillam** is still patiently waiting for an answer to the letter he wrote some time ago, now's the time to pay attention. If you really don't want to put printer control codes into the Quill Printer _dat file, you can insert them into documents instead. One way to do this is to use the decimal code 155, which looks to an Epson-compatible printer like the ESC code which prefaces most printer commands. To put this into a Quill document, you simply hold down the CTRL key and tap the ; key (the semi-colon key). On the screen, this appears as û – a normal u with a 'hat' above it. You will almost certainly have to experiment with this. To avoid conflict with existing Translate entries in your Printer _dat file, take the disk/cartridge containing that file out before printing, to force Quill to use its internal default _dat file. To make sure the printer isn't already storing instructions which you have forgotten about, switch it off and on to reset it.

MEDIA MANAGER SPECIAL EDITION
MEDIA MANAGER

Media Manager Special Edition (MMSE) is a program to be used both when things have gone wrong as well as when things are perfectly OK. It allows for automatic, semi-automatic and manual correction of a huge variety of disk and tape problems. It allows you to explore disks and tapes to your heart's content, producing all sorts of different diagnostic reports. MMSE is very simple to operate, being menu-driven and assuming no degree of computer knowledge whatsoever.

MMSE also allows you to tidy, catalogue, sort and order your disks and cartridges.

The standard Media Manager is both less powerful and less user-friendly, but manages to work on an unexpanded QL.

Both programs allow for data transfer between PC and QL. With MMSE, this transfer is at file and directory level, is bi-directional and is completely automatic.

SPECIAL DESKTOP PUBLISHER
DESKTOP PUBLISHER

These programs are very primitive compared to Professional Publisher. However, if you have not experienced that program as yet, you will find both of these very competent. Both are capable of producing excellent results. The cheaper one has fewer features but is able to run on smaller systems.

EDITOR SPECIAL EDITION
THE EDITOR

With the sole exception of Perfection, this is the best word handling system on the QL. Editor's features include an unrivalled degree of programmability and the ability to cope with the entire 256 character ASCII set. The Special Edition has enhanced document-type facilities, including column blocks and on-screen page break displays. Neither program is suitable for computing novices. Until Perfection, Editor Special Edition would have been our 'Desert Island Program'.

Editor SE can do a few things that Perfection can't, so the ideal combination is to have both (they are compatible at file level and can multitask). If you order Editor SE at the same time as Perfection, you can have Editor SE at half price.

PROFESSIONAL ASTROLOGER
PROFESSIONAL ASTRONOMER

The Astrologer program teaches you Astrology from scratch and enables you to automatically produce text narrative on personality delineation, year-to-year and minute-to-minute life predictions, compatibility interpretations and so on. Whether or not you believe in astrology – indeed, especially if you do not – this program is one that you cannot afford to have. You can tailor the readouts (both in terms of quantity and what is said) to your own particular requirements. The amount of fun you can have with this program is endless. Do not blame us if you start believing in astrology, though!

Astronomer is an extremely fast and accurate solar system calculator, with planetarium views, planet faces, eclipses, cinerama display etc..

TURBO BASIC COMPILER
SUPERCHARGE BASIC COMPILER

Turbo is the finest BASIC compiler for the QL and arguably the finest BASIC compiler for any computer!

Turbo automatically converts working BASIC programs into optimised machine code, with no need for human intervention. The benefits of this conversion are vastly enhanced running speed (as well as much faster loading, encryption and automatic bug fixing for a variety of QL interpreter oddities). Typical speed-up is 20x – 100x.

Turbo is provided with a 200 command toolkit, adding many useful commands to BASIC. Most of these commands will be of immediate use to the programmer, whether he is a novice or an expert. There are commands to load strings and floats into RAM, and to extract them automatically; to search memory and to move its contents; to control jobs and change their priorities...

Supercharge has a few of Turbo's facilities and half its speed.

TOOLKIT III

Toolkit III starts where Toolkit II stopped, adding about 60 new commands and enhancing many existing dual functions. Toolkit III is available either on disk or on ROM, and works whether or not you have Toolkit II.

Toolkit III commands can, with only a couple of exceptions, be compiled using Turbo.

QFLICK CARD INDEX

All QL owners have a copy of Archive, supplied free with the QL. While Archive is competent, it is very hard to get to grips with and is not particularly fast. QFlick presents a very convenient alternative – a snappy , simple-to-use, pointer-controlled card file database. You can move data between QFlick and Archive in either direction.

QFlick is not itself programmable but we document its data structure and give guidance on how to program it using Turbo.

ARCHDEV + RTM
DATABASE ANALYSER
ARCHIVE TUTORIAL
NAMES + ADDRESSES
MAILMERGE
DAT–APPOINT
SEDIT
SCREENPRINT
RECOVER

This suite of utilities will greatly enhance your use of the Archive database system.

Archdev + RTM is a straight replacement for Archive: it gives enhanced speed, greater workspace and a much cleaner boot-up. All your existing applications will work.

Database Analyser provides very fast and comprehensive statistics about your Archive databases.

Archive Tutorial proceeds systematically through the whole philosophy and grammar of Archive, providing you with expert and patient guidance.

Names + addresses, Mailmerge and Dat-Appoint are ready-to-run, off-the-shelf Archive applications, providing an address database, mailmerging and appointment diary respectively. You now have no excuse not to use Archive.

SEdit allows you to create and edit screen format files in Archive. Screenprint allows you to print them out.

Recover allows you to get back lost Archive databases, created when you switched off the computer without properly exiting from Archive.

XREF
SUPERBASIC MONITOR
BETTERBASIC EXPERT SYSTEM

XRef analyses the structure of a BASIC program, providing detailed reports on things like variable usage, what calls what, dynamic call hierarchy of procedures and functions, and so on.

SuperBasic monitor actually monitors and reports on the performance of BASIC programs as they run under the interpreter.

BetterBasic analyses and automatically corrects structural flaws in your programs and allows you to customise things like indentation, number of statements per line, filtering out of noise words, etc.

The three programs together provide a matchless diagnostic and auto-correcting facility for BASIC programs.

TRANSFER UTILITY

This program copies files at high speed between devices, performing translates as it goes along. Ideal for all sorts of applications, including transfers from microdrive to disk.

QMATHS SYSTEM

This is an incredible mathematical compendium for the QL. Pride of place goes to the symbolic problem solver: this can solve equations, simplify expressions, factorise, expand, etc, all symbolically. If you could sneak this one into a maths examination, you would have a formidable ally. QMaths knows about all the algebraic operators, powers, roots, brackets, trigonometry, matrices, determinants, vectors, factorials, permutations, combinations, binomials, exponentials, logarithms, hyperbolics, inverse functions, infinite series including Taylor & Maclaurin expansions, complex numbers, conversions, Fourier series, and lots of calculus: both differential and integral, including integration by parts and definite integrals. QMaths optionally displays its workings and comes with a superb interactive tutorial.

The package also contains an interpretive, fractal, image-generating language with loads of beautiful fractal programs supplied for you to use and edit – no programming skill is required.

There is also a multiple precision floating point maths package, giving calculations at precisions up to over 600 decimal digits of accuracy.

There is even more to this system, but we think we have told you enough.

QMON MACHINE CODE MONITOR

The latest version of Tony Tebby's superb monitor: an absolute must for those who really want to know what is going on in the QL. No other machine code monitor even comes close.

Do not confuse this program with SuperBasic monitor, which monitors SuperBasic, not machine code.

COMPARE

This program compares files – data or program – at colossal speed. Where a mismatch is detected, the relevant areas are highlighted and you can shuffle, displace and align very easily.

CASH TRADER WITH ANALYSER
PAYROLL

Cash trader with Analyser is an accounts system designed by businessmen and not by wretched accountants! Consequently, it has excellent reporting and management facilities, and is very flexible. It is aimed primarily at the layman, probably a sole trader running a small or medium sized business. All the features you would expect – including audit trail – are present.

Payroll is a reasonably flexible system designed to automate the payroll function in small businesses.

Both programs are configurable, with editable defaults letting you adapt the programs from year to year.

HARDBACK WITH FINDER

This is the ultimate hard disk backup and management utility, with all the sophisticated features you could want. User dialogue is via overlapping pop-up windows – the whole program just feels right. It is possible to scan the disk at great speed, too.

DISKTOOL WITH QUICKDISK

This permits you to add password protection to disks, to optionally increase disk storage capacity on DSDD drives by 36K and to increase speed of access by as much as 30%. All this is done while maintaining full compatibility. Automatic file management is also provided.

DIGITAL C SPECIAL EDITION
DIGITAL C

These are extremely fast and efficient C compilers, complying with and surpassing the Small C definition. The Special Edition goes much further, including support for structures, pointers, long pointers, >64K code size, direct access to QDOS traps, etc. The Special Edition C generates code that runs about twice as fast as the other.

THE INFORMATION PAGE

PERFECTION DATA SHEET #2

MENU 1

L	Load file
S	Save file
M	Merge file
K	export blocK
I	export fIle
F	Forward search
B	Backward search
R	Replace
A	replace All
Q	Query the replace
C	toggle Case Sensitivity
E	Emphasise on
O	emphasise Off
U	Underlined on
N	Not underlined
V	moVe block
Y	copY block
D	Delete block
P	Print document
Z	Zap document
X	eXit program

MENU 2

I	Italic on
U	sUperscript on
C	subsCript on
O	italic/superscript/subscript Off
H	Highlight (=strip 2) on
F	highlight oFf
N	back to Normal (strip 1, standard font, no emphasised or underline)
A	reformat pAra
E	reformat from top of para hEreon
G	forced (=hard) paGe break
P	spell Pages
B	spell Block
S	Spell as you type
R	Reinstate line containing cursor
L	cycle end of Line space insert
W	toggle auto line/word Wrap
M	Margins
J	Justification
T	Tabs

MENU 3

T	goto Top of document
B	goto Bottom of document
L	goto specified Line
P	goto specified Page
G	Goto pointer
M	Mark pointer
Z	cycle laZy screen
C	reveal hidden Codes
A	special transpArent insert
H	edit Header
F	edit Footer
D	load printer Data
N	set start (initial) page Number
Q	Quote and edit page length
E	undo current Emphasised region
U	undo current Underlined region
I	undo previous strIp change
S	undo previous Superscript/subscript/italic change
K	toggle Keeping of tabs on reformat

STATUS LINE INDICATORS/FLAGS (DEFAULT = BLANK)
COLumn number LINe number / total lines PAGe number Character code
visible CHARACTERS count hidden/control CHARACTERS count WORDS count
total PAGES LINE number WITHIN PAGE

CAPS	CAPSlock on (Lower case)
NW	No line Wrap (Automatic line wrap at right margin)
CJ/RJ/LJ	Centre/Right/Left Justification (Both L+R justification)
OV	OVerwrite (Insert)
CS	Case-Sensitive search/replace (Case-insensitive)
UP ARROWS	search direction backwards (Forwards)
NQ	No Querying of replaces (Query prior to replacing)
BD	a Block is Defined (No block currently defined)
LZ/VL	LaZy/Very Lazy i.e. accelerated screen-handling (Normal screen)
KT	Keep Tabs always (Replace TABS with spaces when reformatting)
SP	SPellchecking as-you-type (Spell-checking off, or by line/page/file)
0S/1S	insert 0/1 Spaces in EOL region (Insert as many spaces as necessary)

SPELLCHECKER COMMANDS (other than those already in PERFECTION)

1	load 0/1/2 dictionary files (any combination of user and supplied)
2	spellcheck a document file (Perfection or export)
3	exit SPELLCHECKER

DICTIONARY UTILITY COMMANDS

1	extract marked words from file/files/device, add to ASCII wordlist
2	speedsort wordlist (removing duplicates), build user dictionary
3	speedsort wordlist (removing duplicates) for editing in PERFECTION
4	configure defaults for DICTIONARY UTILITY
5	configure defaults for SPELLCHECKER
6	exit DICTIONARY UTILITY

PERFECTION CONFIGURATOR COMMANDS - see sideways-printed list
PERFECTION STRIPSORT COMMANDS - c20 options, no space for details
PERFECTION PRINTER CONFIGURATOR - c50 options inc 16 big translates + 4 strips

PERFECTION CONFIGURATOR OPTIONS (continued)
* startup in overwrite mode? * searching case-sensitive? * search backwards? * auto line-split active? *
auto tab-replace active? * display menu window on startup? * spellcheck to beep on error? * normal
PERFECTION priority * initial keypress repeat delay * subsequent keypress repeat delay * normal time-out for
lazy attribute * quick time-out for lazy attribute * attributes to keep pace with scrolling * maximum
working line length * soft page length * main window background colour * main window paper colour * main
window ink colour * main window border colour * command window background colour * command window paper
colour * command window ink colour * command window border colour * command window error background colour *
command window error paper colour * command window error ink colour * set default tab positions * set sizes
and positions of main window (full/top/bottom) * set window width * set window height * set horizontal
offset of window * set vertical offset of window * centralise main window
Note that these preset defaults may themselves be freely altered within CONFIGURATOR AND FROM PERFECTION.

DIGITAL PRECISION LTD

STOP PRESS/PRODUCT INFO

We are on the look-out for superb QL programs: our aim, to make them even better and then include them in our range. Write if interested.

Please see the outside back cover for more information on PERFECTION and on other Digital Precision gear. Here's what PERFECTION does:

DIRECT KEYPRESS COMMANDS

CTRL/	CTRL/ALT/
A cycle justification setting	immediate replace All
B goto Bottom of document	Backward search
C cycle job (QDOS)	Copy predefined block
D Define highlight as block	Delete predefined block
E Emphasised (bold) on	Emphasised (bold) off
F Forward search	Footer edit
G immediate search in direction	Goto line number
H reveal Hidden codes	Header edit
I insert tab (QDOS)	set Indent margin here
J enter (QDOS)	recall last entry (TK2)
K insert hard=forced page break	export predefined block
L Load file(native/Quill/export)	set Left margin here
M Merge exported file	Move predefined block
N back to Normal (reset)	spellcheck pages
O italics/super/subscript Off	spellcheck block
P Print document to printer/file	goto Page number
Q Quote / edit page length	toggle Query before replace
R Reformat this paragraph	set Right margin here
S Save file	Spellcheck as you type
T goto Top of document	set Tab positions
U Underlined on	Underlined off
V set replace string	set initial page number
W put replace string here	toggle word/line Wrap setting
X immediate replace in direction	exit PERFECTION
Y italics on	toggle case sensitivity
Z undefine current block	Zap (i.e. clear) document

TAB	insert TAB
ENTER	mark end of paragraph
CTRL/POUND	special transparent insert
CTRL/SHIFT/\	split very long lines
CTRL/SHIFT/]	cycle EOL space insert
CTRL/SHIFT/POUND	cycle lazy screen
CTRL/SHIFT/ESC	wait for keypress
ALT/CAPS	communicate with SPELLCHECKER
CTRL/CAPS	communicate with PROPUB_INSERTER (picture inclusion)
CTRL/ALT/CAPS	load printer data
SHIFT/CAPS	reformat paragraph from here
SHIFT/ALT/CAPS	reformat hereon down through whole document
CTRL/SHIFT/CAPS	load macro from predefined block
CTRL/SHIFT/ALT/CAPS	export file
CTRL/F1	remove the emphasis of this region
CTRL/SHIFT/F1	undo previous italics/superscript/subscript change
CTRL/F2	remove the underlining of this region
CTRL/SHIFT/F2	undo previous strip change
CTRL/F3	execute macro
CTRL/SHIFT/F3	load macro file
CTRL/F4	superscript on
CTRL/SHIFT/F4	subscript on
F5	clear previous command prompt
CTRL/SHIFT/F5	display alternate status line
SHIFT/TAB	toggle keep/replace tabs
SHIFT/ENTER	undo changes to current line (reinstate)
SHIFT/ALT/1-8	mark Pointer 1-8
ALT/1-8	goto Pointer 1-8
ALT/F1-F4	set strip 1-4 (1=normal, 2=highlight)
SHIFT/ALT/+	set search direction to forward
ALT/-	set search direction to backward

All other ALT keys available for customisation: use the MENU option where a direct PERFECTION keypress clashes with a non-configurable TASKMASTER key.

	SHIFT	CTRL	ALT	CTRL/SHIFT	SHIFT/ALT	CTRL/ALT	CTRL/SHIFT/ALT
UP /DOWN	by line	by para	cursor static	by page	alter case of word	by screen depth	goto top or bottom	delete line cursor up/down
RIGHT /LEFT	by char	by word	delete char	to EOL	delete word	by screen width	search for highlight	delete to EOL right or left

INDIRECT (i.e. MENU) COMMANDS

TOP (DEFAULT) MENU

F1	Toggle Help
F2	Toggle menu visibility
F3	cycle menu 1-2-3-1
F4	Refresh screen
F5	Switch between windows
SHIFT/F1	Re-perform last command
SHIFT/F2	Toggle L/R menu visibility in Mode 8
SHIFT/F3	cycle menu 3-2-1-3
SHIFT/F4	Resize/move windows
SHIFT/F5	Toggle Single/dual window mode
ESC	Cancel command, escape from sequence
ALT/F5	Toggle Overwrite/Insert

PERFECTION DATA SHEET #1

DIGITAL PRECISION LTD

PERFECTION CONFIGURATOR OPTIONS (i.e. PRESET DEFAULTS)
Screen Sizes and Positions * Screen Colours * PERFECTION Cleanup Priorities * File and Device Names * Font Names * Maximum Line and Page Lengths * Autorepeat Delays * Lazy Screen Parameters * Tab Positions * Miscellaneous Parameters * Justification and Margins * PERFECTION Job Priority * Emphasised Ink Colour and Strip Colours * Action when Appending to Lines * Baud Rate * Exit * Update PERFECTION? * ink colour for emphasised * highlight strip 2 colour * alter strip 3 colour * default end of line action 0/1/S * startup macro file name * current printer device or file name * current printer data device and file name * current help device and file name * current default device and/or file name * current default font device name * current name for the standard font * current name for the superscript font * current name for the subscript font * current name for the italic font * minimum priority for PERFECTION cleanup job * maximum priority for PERFECTION cleanup job * default justification * left hand margin * right hand margin * indent margin * suppress lazy screen? * suppress very lazy screen acceleration? * lazy screen acceleration factor .....contd

SPECIAL DEALS

5% off total if you buy 2 programs/upgrades;
10% off 3; 15% off 4; 20% off 5; 25% off 6+
Upgrades cost difference in price + £10
Non-UK Europe add 5%, rest of world 10%

For full terms and conditions, please refer to any of our QL World ads from Jan-Nov 1990, or write in including a SAE

CPORT BASIC TO C CONVERTER

This program translates SuperBasic programs directly into C source code, automatically. This C source code may then be edited or compiled. If you want to move programs to C for migration to other hardware, or want to accelerate your programs, or just want to learn C the easy way (chuck BASIC in one end and examine the C that spews out of the other), CPort is the system for you.

CPort is friendly and tolerant of poorly written BASIC. There is even a method of dealing with unusual BASIC keywords. The generated C, which can be switched between the ANSI and Lattice industry standards, is very readable and is often optimal. CPort's user interface is extremely friendly. CPort is available with or without the C68 compiler.

SUPERFORTH COMPILER WITH REVERSI

Forth is the most logical computer language. This compiler produces multitasking code. The manual teaches you Forth-83 from scratch.

IDIS SPECIAL EDITION
IDIS

These intelligent disassemblers make the otherwise terrifyingly complex task of understanding other people's machine code programs absurdly easy. The SE version, which has a higher hardware requirement, sorts out some routines, replaces addresses with names, untangles data from code and much more.

QKICK FRONT END SYSTEM

This is a simple, easy-to-master, pull-down menu controlled multitasking front end. QKick runs in the background and can be called up at any time. It provides you with notepads, sophisticated file/sector/RAM handling, backing up facilities, a clock, diary, calculator, mini-database and so on.

ADVENTURE CREATION TOOL SPECIAL EDITION

ACT is a must for every programmer. The name of the program is misleading, insofar as it has capabilities far beyond the 'mere' creation of adventures. ACT has utilities providing animated graphics, data compression, language design, parsing, maps, object-oriented control etc. If all you want to do is generate adventures, though, you do not need to be a programmer to use it. This is a purchase you will never regret.

PEDIT

A fast, modern and capable printer driver for the programs bundled with the QL.

MICROBRIDGE

Superb contract bridge bidder (ACOL etc) and player, using millions of random but reconstructable hands. Microbridge also includes a state of the art interactive bidding tutor and a clear instruction manual. There is nothing like this anywhere else!

SUPER ASTROLOGER

A very cut-down version of Professional Astrologer – still great fun, though!

SUCCESS CP/M EMULATOR

Allows your QL to run CP/M programs at great speed.

£			
169.95	eT	PC CONQUEROR WITH DR-DOS v5.0
129.95	dT	PERFECTION PLUS WITH SPELL CHECKER
119.95	eT	THE SOLUTION WITH DR-DOS v5.0
99.95	cT	CPORT WITH C68K COMPILER
99.95	cT	CASH TRADER v3.0 + ANALYSER
89.95	dT	PERFECTION
89.95	eT	PC CONQUEROR
89.95	cT	PROFESSIONAL PUBLISHER
89.95	cT	CPORT BASIC TO C TRANSLATION SYSTEM
79.95	aT	TURBO BASIC COMPILER + TOOLKIT
69.95	aT	PROFESSIONAL ASTROLOGER WITH ASTRONOMER
59.95	cT	QMATHS MATHEMATICAL SYSTEM
59.95	aT	PROFESSIONAL ASTROLOGER
49.95	+aT	LIGHTNING SPECIAL EDITION
49.95	eT	PROFESSIONAL PUBLISHER TOOLBOXES (PAIR)
49.95	eT	ACT SPECIAL EDITION
49.95	dT	THE EDITOR SPECIAL EDITION
49.95	dT	SPELLCHECKER
49.95	dT	MEDIA MANAGER SPECIAL EDITION
49.95	dT	3-D PRECISION CAD SYSTEM
49.95	cT	PAYROLL
49.95	cT	HARDBACK + FINDER
49.95	bT	SUCCESS
49.95	aT	DIGITAL C SPECIAL EDITION
44.95	+a	TOOLKIT III WITH ROM
39.95	eT	THE SOLUTION
39.95	dT	IDIS SPECIAL EDITION
39.95	ct	SPECIAL DESKTOP PUBLISHER
39.95	at	QMON MACHINE CODE MONITOR v2.05
39.95	aT	SUPERFORTH COMPILER WITH REVERSI
39.95	aT	EYE-Q
34.95	d	MICROBRIDGE
29.95	gT	MEGA DICTIONARY
29.95	fT	PROFESSIONAL ASTRONOMER
29.95	eT	PROFESSIONAL PUBLISHER TOOLBOX
29.95	eT	PROFESSIONAL PUBLISHER TOOLBOX PART TWO
29.95	a	TOOLKIT III
29.95	a	SUPER SPRITE GENERATOR
29.95	a	SUPERCHARGE SPECIAL EDITION
29.95	a	QFLICK CARD INDEX SYSTEM
29.95	a	PERFECT POINTER TOOLS
29.95	a	GAMES COMPENDIUM (ALL FIVE GAMES)
29.95	at	THE EDITOR
29.95	at	DIGITAL C COMPILER
29.95	aT	XREF v2.0
29.95	aT	SEDIT
29.95	aT	ARCHDEV + RUN-TIME MODULE
24.95	f	SUPER ASTROLOGER
24.95	d	DESKTOP PUBLISHER
24.95	a	QKICK MULTITASKING SYSTEM
24.95	a	MEDIA MANAGER
24.95	at	IDIS INTELLIGENT DISASSEMBLER
24.95	aT	SUPERBASIC MONITOR
24.95	aT	LIGHTNING
24.95	aT	BETTER BASIC EXPERT SYSTEM
19.95	eT	FONT ENLARGER
19.95	dT	NAMES AND ADDRESSES
19.95	dT	DAT-APPOINT
19.95	b	DISKTOOL WITH QUICKDISK
19.95	aT	ULTRAPRINT
19.95	aT	SCREENPRINT
19.95	aT	RECOVER
19.95	aT	PEDIT
19.95	aT	MAILMERGE
19.95	aT	DATABASE ANALYSER
19.95	aT	COMPARE
19.95	aT	ARCHIVE TUTORIAL
9.95	f	SUPER BACKGAMMON GAME
9.95	f	REVERSI GAME
9.95	f	DROIDZONE GAME
9.95	f	BLOCKLANDS GAME
9.95	f	ARCADIA GAME
9.95	eT	GRAFIX
9.95	b	TRANSFER UTILITY

NEW!

* KEY: + As well as mdv disk, you get a ROM
a Available either on mdv or disk
b Available only on disk
c Minimum 512K RAM:only available on disk
d Minimum 256K RAM:either mdv or disk
e Minimum 256K RAM:only available on disk
f Available only on mdv
g Minimum 1.5M RAM:only available on disk
T Compatible with THORs including XVI
t Compatible with all THORs except XVI

Send to: **DIGITAL PRECISION LTD, 222 THE AVENUE, CHINGFORD, LONDON E4 9SE** or phone **081-527 5493**

Please rush me: .....

Name: ..... Address: .....

..... Postcode: .....

Encl: CHEQUE/VISA/ACCESS/MASTERCARD/PO/MO/CASH for £ .....

Card No: ..... Expiry:..... Signed: .....

DIGITAL PRECISION LTD

Delete as appropriate: (QL/THOR/STQL)(MDV/3.5"/5.25")(Disk=360/720/1440/2880/6400 sectors; nb 1 sector=0.5K)(If 720 sectors: SSDD/DSSD)

NOTICE BOARD

QL MAGIC

When a friend of mine is gritting his teeth over his QL, and I ask him what I can get him to ease the pain, he says "get me some magic".

What he means, of course, is that he wants the program, or the printer, or whatever is bugging him to stop hurting and start working.

Magic is a very handy thing if you can get it. That is why I am thinking of calling a new QL World column 'QL Magic'. It won't help you turn your least favourite niece or nephew into a frog, but it may help you to turn a pig of a program into a useful item, or get something moving which has been doing a Garfield despite your best persuasions.

So we will be printing short practical suggestions and programs which people have found helpful in unsticking the chewing gum in the works of computer life (even a remedy for removing real chewing gum, should somebody come up with a good one. Don't ask us to test it, though!).

If you have any useful tips or routines, send them in. See below for a guide to laying out articles.

EXPERT HELP

The good news is: an eminent member of the QL hardware community has promised us to have a crack at some of the knottier hardware queries we receive. He is, like most of the expert hands in the QL business, an immensely busy person, so his offer is very much appreciated. If you have queries to shed, prepare to shed them now. Help us, and keep them as concise as you can. One query per letter, please, to keep things simple.

GUIDE FOR AUTHORS

By popular request, here are some guidelines for readers sending in articles for possible publication:

We take TEXT in the good old-fashioned way, on paper, double-line-spaced, one side only. Leave at least an inch of margin all round. We write our own sub-heads, so only include them as a rough guide. Put your name and address at the start of the article. LISTINGS, on the other hand, must be single-spaced and printed to a width which will allow us to print them to a reasonable size within (usually) half the width of our page. Use your judgement, according to your printer type. They must be on separate white sheets, and dark enough to reproduce clearly.

Don't send disks or mdvs unless we request them.

We PAY around £60 a printed page, varying slightly with circumstances. We don't pay for news items unless the writer is unconnected with the matter being reported, and jolly quick off the mark. Such items are always bylined (and rare).

We don't pay for Open Channel, of course, but we don't insist you double-space it (or even type it) either.

For a more detailed factsheet, send an SAE to the Editor.

Z 88

EaZy Money	£71.00	Centronics Interface	£22.25
Z-Base	£71.00	128K Eprom Pak	£28.00
Z-Tape	£51.50	128K RAM Pak	£38.75
Z-Term	£51.50	Z88 Computer	£223.00
ThinkZ	£51.50	512K Ram Pack	£103.50
Superchip 4	£154.00	256K Eprom Pack	£68.25
Golf	£25.50	1Mgb Ram Pack	£220.00
Topper	£10.50	Squeeze 88	£47.00
		Superchip 3	£133.50
		Z88 QL Link	£27.50

PSION SERIES 3

128k Computer	£200.00	256k Computer	£250.00
128k Flash card	£40.00	256k Flash card	£70.00
512k Flash card	£120.00	1MgB Flash card	£200.00
2MgB Flash card	£300.00	128k Ram card	£80.00
512k Ram card	£200.00	1MgB Ram card	£300.00
Parallel interface	£30.00	Mains adaptor	£15.00
PC link	£70.00	Mac link	£70.00

TK COMPUTERWARE

the QL stockist

Tel: 0303-81-2801

PSION ORGANISER 2

Model LZ	£100.00
Model LZ 64	£130.00
Comms Link	£60.00
Comms Link – Mac	£73.00
Travel Pak	£40.00
Harvester Superchip	£100.00
Letter Organiser	£40.00
Informed Organiser	£40.00
Diary Link	£13.00
128k Datapak	£100.00
64k Datapak	£60.00
32k Rampak	£60.00

SPARES

Keyboard Membrane	£8.50
JS ROM set	£31.00
ZX 8301	£15.50
ULA ZX8302	£10.50
MAB 8049	£7.50
QL Power Supply	£31.00
MC 68008	£26.50
M/drive assembly-(complete)	28.00
Microdrive Assembly (exchange)	£21.50
Voltage regulator (high output)	£5.00
QPower Regulator	£25.50
QL Test kit	£14.50
Repairs	phone

SUNDRIES

VDU cleaning kit	£7.50
Disk Cleaning Kit 3.5"	£15.50
10 Microdrive cartridges	£30.00
100 Microdrive labels	£4.00
Discs 3.5" 10 DSDD (branded)	£26.00
Power Clean Mains Filter	£24.50
Psion 2.35 Set	£23.50
Printer Ribbons	Phone
Tandata QCOM set	£90.00
QUIMI I/F with mouse	£101.50

SOFTWARE

BUSINESS

Project Planner	£30.00
Sales Invoice	£20.50
Small Traders Pack	£30.00
Stock Accounting System	£41.00

TEXT

Master Spy	£31.00
Page Designer 2	£36.00
Perfection	£90.00
Perfection +	£130.00
Professional Publisher	£90.00
QTYP 2	£31.00
SPY	£15.50
Turbo Quill Plus	£14.00
The Editor	£30.00
The Editor Special Edition	£50.00

FINANCE

Home Finance [Buzz]	£22.50
Investment Monitor	20.50

CAD

Eye Q	£40.00
GraphicQL Plus	£25.50
PCB 1	£102.50
PCB 2	£51.50
TechniQL	£51.50
Technikit	£25.50
The Painter	£51.50
3-D Designer	£39.00
3D Precision	£50.00

PROGRAMMING

Assembly Workbench	£25.50
Basic ally	£20.50
Better Basic	£25.00
Digital C	£30.00
Digital C Special Edition	£50.00
Forth	£31.00
IDIS	£25.00
IDIS Special Edition	£40.00
Macro Assembler - GST	£39.00
Macro Assembler - Talent +	£20.50
Pro Fortran - 77	£101.50
Pro Pascal	£101.50
QMON 2	£20.50
Lightning Special Edition	£60.00
Locksmith	£15.50
Mailfile	£15.50
Media Manager	£25.00
Media Manager Special Edn	£50.00
QFlash Ramdisc + t'kit	£19.50
QFlash Ram Eprom	£25.50
QFlash Ram Utilites	£10.50
QPAC 1	£23.50
QPAC 2	£40.00
Q Top	£31.00
RPM	£15.50
Sidewinder Deluxe	£20.50
Super Toolkit 2 (Eprom)	£30.00
Super Toolkit 2 – Reconfigurable	£30.00
Success	£50.00
Terminal	£31.00
Touch Typist	£12.50
Ultraprint	£20.00
XRef 2.00	£20.50
4 Matter	£15.50

LEISURE

Bridge Player 2	£15.50
Cosmos+	£15.50
Flight Simulator	£19.50
Matchpoint	£15.50
Micro Bridge	£40.00
Prof Astrologer + Astronomer	£80.00
Prof. Astronomer	£40.00
Snooker	£13.50
Strip Poker	£15.50
Super Astrologer	£15.00
Super Backgammon 3	£13.50
Type 22	£18.50
War in East	£35.00

GAMES

Aquanaught 471	£19.50
Arcanoid	£15.50
Stone Raider	£13.50
Brain Smasher	£18.50
Crazy Painter	£13.50
Deathstrike	£15.50
Firebirds	£15.50
Hopper	£13.50
Hoverzone	£15.50
Jungle Eddy	£15.50
King	£13.50
Land of Havoc	£19.50
Lost Pharoah	£15.50
Stone Raider	£13.50
Super Arcadia	£10.50
Cuthbert in Space	£13.50

UTILITIES

Archivist	£31.00
Archivist MP	£41.00
ArchRTM+ArchDEV	£20.50
Cartridge Doctor	£18.50
Conqueror MS-DOS 4.01	£125.00
Conqueror with DR-DOS 5.00	£180.00
Discover	£31.00
Lightening	£30.00
Super Forth + Reversi	£40.00
Super Sprite Generator	£31.00
Supercharge Special Edition	£25.00
Turbo	£80.00

• Please telephone for details of other products not listed here •

BOOKS

Assembly Language Prog	£9.00	Using Your Z88	£10.00
Z88 Dabhand Guide	£16.00	Z88 Computing	£10.00
Z88 Ref Manual	£16.00	Z88 Software DLV Guide	£40.00
Z88 Magic	£16.00	Using Psion Organiser	£10.00
Z88 Pipedream	£16.00	Psion Handling Book	£15.00

ALL PRICES INCLUDE VAT. U.K. POSTAGE £2.50 PER ORDER. - WE EXPORT WORLDWIDE

Phone for details of other QL products OR send SAE for our latest catalogue.

Credit card phone orders accepted, or send cheques, postal orders, Eurocheques, to:

TK COMPUTERWARE, STONE STREET, NORTH STANFORD, ASHFORD, KENT TN25 6DF

TELEX 966676 PMFAB G

VISA · Access · MasterCard · AMERICAN EXPRESS · EUROCARD

FAX 0303 812892

QL SCENE

DataDesign and The Painter get a remodel

Progs Van Auwera have updated two of their longstanding product range. First off, *Datadesign* has been extensively revised so that programing it via SuperBasic is 'much easier'. The SuperBasic interface has been considerably simplified; the old interface can still be used by those who prefer it and have invested time in it, though. Extra commands have been added; MERGE allows the program to merge files almost as fast as it loads them; FIND and FILTER are improved. More information is available from Progs Van Auwera.

The Painter has also been updated. It will work with the cursor keys on the Gold Card; there are new printer drivers for the HP DeskJet and other printers which use to HPGL graphics language, for both 150 and 300 dpi print resolutions.

Current prices of the major programs in Belgian francs and Sterling are: The Painter 4.04, BEF 1500 (£25); The Clipart BEF 600 (£10); Qractal BEF 1200 (£20); Datadesign 2.05 BEF 3000 (£50).

For other prices and upgrade prices and conditions, contact **Progs Van Auwera at Haachtstraat 92, B-3020 Veltem, Belgium. Tel: 010 32 16 488952.**

Third time cute

"Compare these speeds' say Digital Precision about *Perfection* 3.00 – instant navigation, searching approximately 100 pages (100K) per second, block handling about 20 pages per second, on a standard QL. 'Four to five times faster still' with a Gold Card. 'Who needs Perfection?' asks a little man in the corner. 'Perhaps you need Perfection', say Digital Precision. Why is the little man wearing baggy spotted trousers, a funny hat and a leaky handbag (or is it a bucket of water)? Does he appreciate cute menus and good coding? Who is AT, anyway? Perfection 3.00, £79.95, or £119.95 with the Spellchecker, from Digital Precision, 222 The Avenue, Chingford, London E4 9SE.

Mouse in MUG seen in Merseyside

QL MUG (Merseyside User Group) has contacted QL World to say that they are still flourishing. Formerly a Quanta subgroup and still affiliated, QL MUG is open to all QL users in the area. They also design and produce (in kit or ready-built form, or parts list, and points in between) the Merseyside Mouse for the QL, which has been building its market in 1991. Contact Mr James at 3 Barnard Road, Birkenhead, L43 1TT. Tel: 051 652 7366.

OPEN CHANNEL

Open Channel is where you have the opportunity to voice your opinions in *Sinclair QL World*. Whether you want to ask for help with a technical problem, provide somebody with the answer, or just sound off about something which bothers you, write to: Open Channel, Sinclair QL World, 116/120 Goswell Road, London EC1V 7QD.

Superior

I always turn to your Notebook for up to date information and a little light entertainment on receiving *QL World*. Your January effort was unusually doleful, however. The literary quality of the magazine is far superior to other journals with a technical or specialist content.

In reply to the cryptic remark in your penultimate paragraph, the top half of page 39 in the December issue is a reprint of page 46 of the September issue. Also, we have too many 'm's in systematic on the Contents page, and on the Noticeboard page the paragraph heading C COMPILER has its P missing. The heading of the article on page 36 has 'questionnaire' misspelt twice with only one 'n'.

However, this gives me an excuse to write and wish you every success in overcoming any difficulties you may have in publishing, printing and distributing the magazine in these difficult times. I am too old to learn another language, and wish to continue using the QL.

Wishing you and your staff a peaceful and prosperous 1992.

Douglas D Perry
Walsall
W Midlands

Editor's comment: What a nice letter. Christmas passed without anybody upending anything over anyone, although there were a few stern words spoken at home about how many days' work we could fit in between 9pm on Thursday and 4pm on Friday. Just returned from a much-needed week's holiday in the snow, I am greatly cheered up; also by knowing that whatever went wrong on January or February, it can't match December; also by this letter. No banana on the half-page, though. The first caller was Bryan Davies, wanting to know if he gets paid again for it. We said, not unless he can explain how it got there. Then there was the conversation about questionnaire, with the OED being produced to prove that you can spell it with one 'n', and various proofs on my behalf to demonstrate: that's all very well, but I asked for two. Words were spoken, but not too many, however; the setters are just as short of hands and eyes as we are. The extra 'm' in systematic I achieved all by myself, without any help; it was certainly noticed (too late, as usual). The mystery boob, however, remains mysterious.

Editor's notebook

On the back of these very words is a £1-off entrance voucher to the March All Formats Computer Fair of your choice. If you can't bear to part with my words, copy them down by hand into your anthology, and you can clip the coupon with a clear conscience.

Flap flap oink . . .

OK. At least, first note well that the Sinclair QL User Club eV in Germany is holding the International QL Meeting in Munster on 21 March 1992, and it will be very well attended by suppliers and clubs, by the look of it. Quanta is sending a minibus. Enquiries c/o Andreas Rudolf, Gruner Weg 25, W-5202 Hennet 1, Germany. There's a long list of workshops.

The spot-the-grolley 'competition' has rounded up the usual suspects, and one innocent party, but has yet to read the state of an Editor's mind as press day approaches. To confuse the issue thoroughly, last month's brainstorm seems to have produced the opposite result.

One last thing: rumour has it that we're going to get a Gold Card for review soon

Scrabble

I am a Scrabble fan, and possessed the version as marketed by Leisure Genius for the QL. Unfortunately I have corrupted my copy, because I neglected to remove the cartridge and clone from the machine before switch-on. As a result the microdrives whirled into action and corrupted the main Scrabble cassette on mdv 2.

To my dismay I find that Leisure Genius no longer trades, and the game is no longer available. Since I'm unemployed I really enjoyed a late night game in preference to watching tv! Could you advise me how I can obtain this game for the QL?

Barry Hammett
Swansea

Editor's comment: Nobody is listing this game at the moment. If anyone can help, please write to us and we will pass the offer on.

Port connectors

There have been several queries recently concerning the connectors used for the RS232 ports and Control ports – many people seem to think these connectors are not freely available. This is not the case.

These connectors are produced by Bicc-Vero Electronics Ltd, and are sold through the STC group of companies. They are available in any colour you like as long as it's black. There are four different types available: four way left or right handed, and six-way left or right handed. The QL uses six-way left handed for the Control ports and six-way right-handed for the RS232 ports. The Bicc-Vero part numbers for these plugs are 900-71053C and 900-71088J.

A wire-stripping tool is available under part number 900-71019A and a termination tool under part number 900-71020B.

The connectors can be fitted without these tools or by using tools supplied with PTC telephone plugs.

STC are currently selling these plugs at just under seven pence each *(£0.659 is the price quoted, although you may have to buy in bulk or have an account to take advantage of this price - Editor.)*, but if you are not in the trade you are probably better off contacting ITT-Cannon who now own Bicc-Vero, on 0256 473171 and asking for your nearest public stockist.

If all else fails, your local tv/radio repair shop may be prepared to order the parts for you when ordering their own stock, but be prepared to wait, and to pay a bit over the odds for this type of service.

I hope this information helps the people who are still laboriously filing down telecom plugs and those who have not yet experienced that delight. It's a pity these connectors never caught on in a big way, as they are quite good little beasts; it seems that only Sinclair and British Telecom were impressed enough to use them in any quantity.

Terry Griffiths
Stevenage
Herts

Add-ons

I am a relatively new QL user living in Greece. I must confess that my knowledge of Qdos is very limited, but I think I know the hardware structure of the QL well enough to suggest some feasible but dramatic add-ons.

Reading *QL World* every month, I often find users complaining about the screen display. Every user would like a better and faster display. However, any attempt to improve the already-existing video circuitry on the QL is bound to end in misery.

The solution is a second video circuit accompanied by appropriate software drivers. The only thing needed is some free address space and a company to undertake the enterprise. Free address space may not exist in the normal QLs where the 68008 can address up to 1MB of memory, but it does in the Gold Card expanded QLs; in fact, the 68000 can address up to 16MB, so 14 MB remains unused. So we have address space enough to host an XVGA display.

But there is no need to design a totally new video circuit. Suppose we could emulate the basic signals of a PC slot (AEN IORD MEMRD MEMWR CLK RESET data and address bus), and rearrange them in the right order, then we would be able to fit in any PC video display cards (which only use the above signals). Does that sound difficult? It isn't, and I mean that. By the time you read this letter I hope I will be using a real PC Hercules card on my little 48K Spectrum (I dare not touch my QL, nor have I the necessary experience yet).

I think such an emulator would be a piece of cake for experienced QL hardware people. The only obstacle I can think of is the absence of an expansion port on the Gold Card (which is essential for free address space), but we shouldn't get stuck there.

Even if we take as a fact that there is no easy way to use free address space (at least 64KB for an EGA display and at least 1 MB for a Super VGA) on simple or Gold Carded QLs, I can think of another way to create a perfect display for the QL. This time only 256 bytes are required which I suppose would be easier (to control a screen up to 2048 x 2048 x 256 colours).

This time the cpu does not see directly the video ram, but instead uses 256 addresses to issue commands on a graphic processor which in its turn addresses directly the video ram which can be up to eight cpu-independent MBs. The graphic processor I have in mind is the SGS-Thompson TS68483, which is the best of the series GPs (graphic processors) in my opinion.

In such a system the video ram is roughly divided into two parts, the active video mem and the notepad video mem. The active video mem is projected on the screen. In the notepad mem, we store certain shapes such as characters, music symbols and even alien faces; then we tell the GP to move the required shape from notepad to active mem – that's all. With this specific GP, we can issue commands to draw lines, arcs, polylines, polyarcs, filled or otherwise, to move blocks with mask or not, and many other things.

Needless to say, all these actions require almost no cpu time, and are executed at speeds that even *four* 80486s at 33 MHz cannot achieve (two megadots per second). Finally, the shapes are stored in the notepad mem, not in the usual way (the cpu cannot see the memory). The cpu issues a LOAD command to the GP, and then feeds it byte by byte the shape which is stored in a window originated by the parameters of the LOAD command.

In a similar way, we can save a window of active (or otherwise) memory from GP to the cpu. Anyone looking for some detailed information should refer to the ST SGS_Thompson Graphic Processors Databook.

A system like that would be a real solution for cad, simulators, virtual reality programs, etc. and would allow software companies to write the best programs ever written on a home machine.

I hope I've given someone some ideas, I would like to hear other people's opinions.

Stavros Grigorakakis
Athens
Greece

QLipboard

I thought I'd just point out an oversight in respect of an item in *Troubleshooter* in November's *QL World*. Under the heading Cut and Paste, you describe almost verbatim the facilities available as part of Simon Goodwin's *DIY Toolkit* package QLipboard! This appeared in the January and February 1991 issues of QL World. It does not, of course, provide a DDE facility nor cater for the transfer of graphics but, where text is concerned, it is more than adequate, being capable of recognising different fonts and character sizes (although not automatically).

The package is probably best obtained from Richard Alexander at CGH Services if you don't want to type it in. It is not a product of polished commercial standing, unsurprisingly. The user interface is somewhat esoteric and complex. Also, text is clipped and copied line by line and not as Bryan suggests (and I would like to have seen) on a cursor-start/cursor-end basis. But in other areas it goes much further, with up to ten 'pages' to clip text into and a full editor to edit the clipped text before it is pasted. With possibly a bit more development, Simon could have produced a product to rival his *Speedscreen*.

If you are a *Qpac 2* user, then a conflict exists with the very nature of QLipboard's operations, namely its need to access the windows of other tasks. This is contrary to the 'locked' windows concept of Qpac 2, that is, you have to 'unlock' the windows of those tasks with which QLipboard is to be used. I have no experience of how QLipboard operates with Taskmaster or other multi-tasking environments.

Stephen Mitchell
Walton-on-Thames
Surrey

Appellation

Re: Missing magazine
I apologize for not addressing you correctly, but I could not make out your name from your signature. This reply is to thank you for locating a March issue of *QL World*. I am also impressed by the joviality of your letter. It's good to see people happy in their work.

Brent Hayhoe
New Barnet

Editor's comment: It must be the raspberry-leaf tea again. In view of some recent correspondence about computer manuals, it's interesting to note that much useful information about QL World appears on the left hand side of the first (Contents) page of QL World, including our address, the address of our subscription agents, current subscription prices, and so on. We get a fair number of letters asking for this information. My name appears with that of many other toiling minions at the top left, under the appellation 'Editor', but after a decade of trying to convert the public, I now find the traditional forms 'Dear Editor' or 'Dear Sir' perfectly acceptable. You have to bow to demand.

PSION WAVES

Nigel Bates takes a headlong plunge into biorhythms through the medium of Psion Easel - can it see into the future?

Have you ever wanted to make more use out of Psion *Easel* graphics software than merely plotting your monthly income and expenditure or tabulating yearly rainfall statistics for the British Isles?

The following is an Easel application which plots biorhythm curves for any period, past or future, and requires only three formulae to be typed in. It also explains some simple mathematics and once again demonstrates the impressive power of the Psion software.

Biorhythms are curves showing sinusoidal patterns of physical, emotional, and intellectual energy on a daily basis, and are claimed to be a guide to periods of both well-being and stress in all three spheres. They arose out of the research of Dr. Wilhelm Fliess in Berlin at the beginning of this century, later supplemented by Alfred Telscher, an Austrian professor in the 1920s. Beginning at birth, biorhythms oscillate with regular periods of 23, 28, and 33 days respectively. In examining the curves, it is important not only to note the days when they are at their maximum or minimum values but also on which days they cross from positive to negative or *vice versa*. These are known as critical days and are the times when you are most likely to be on edge and make mistakes and wrong decisions.

On your cycle

To know at which part of each cycle you currently are, you need to know two things: your age in days and how many complete cycles have already elapsed. The first can be worked out with a calculator, taking account of leap years and the fragments of years for the current year and the one in which you were born. A much simpler, and readily available, method is to use the days( ) function in *Archive* or *Abacus*. Let us assume today's date is 1st July 1991 and your birthday is 23rd May 1955, simply load Archive and type in:

```
print days("1991/07/01") - days("1955/05/23")
```

In Abacus, omit the word 'print'. Pressing Enter gives the result 13,188 days. It is essential to include both your birth date and the target date, so add 1 to the above result to give 13,189 days. You could also use SuperBasic, but only if you were born after 1st January 1961, since SuperBasic's DATE function only returns correct values for dates after this. For example, if your birth date is 29th February 1968 enter the following short program:

```
100 SDATE 1991,7,1,0,0,0 : today=DATE
110 SDATE 1968,2,29,0,0,0 : birthdate=DATE
120 PRINT (today - birthdate)/86400 + 1
130 SDATE 1991,7,1,0,0,0 :rem reset the clock to whatever is the correct time
```

If you have *Minerva* then you can enter the following directly:

```
print (DATE(1991,7,1,0,0,0) - DATE(1968,2,29,0,0,0))/86400 + 1
```

where the value in seconds returned by the DATE function is divided by 86400 to convert it to days and 1 added as explained above to give the result 8,524 days.

We now have all the information we need to construct the biorhythm curves in Easel. The *QL User Guide* tells us that a sine wave can be drawn in Easel by using the formula:

```
sinewave = sin(2*pi( )*(cell-1)/(cellmax-1))
```

where 'cell' is the name of a set of figures always in Easel which has the value of the cell number eg its value at Jan is 1, at Feb is 2 and so on. Cellmax is a reserved word with a value equal to the number of cells currently displayed on the screen, and so is useful for scaling expressions along the label axis. Thus the above formula will draw one cycle of a sine wave irrespective of the number of cells on the screen. A point to remember is that if you increase the number of cells after the sine wave is drawn, by pressing the TAB key to move the cursor off the right-hand edge of the screen, Easel will not extend the curve into the new area of the screen. You need to type the formula in again to do this.

There is a problem in that we really want to have three biorhythm sine waves, with their different periods, on the screen simultaneously, and so we cannot use cellmax. The sine function, like all the trigonometric functions, is periodic and repeats after 360 degrees or 2 radians. So the formula:

```
Physical-sin(2*pi( )*cell/23)
```

will draw a sine curve, representing your physical biorhythm, with the correct period. If we take the first cell of the graph to be today's date, then, as it stands, the curve drawn by the above formula will be your correct physical biorhythm only if your age (as returned by the days( ) function described above) is an exact multiple of 23 days. It would be tedious to have to wait for the next such coincidence every time you wanted to get an up-to-date chart. To get the right proportion of the biorhythm on the screen, we need to introduce the concept of *phase*. This is the amount by which the sine wave is shifted along the axis from the origin according to the number of days you are into your current cycle. To find this out we need what is known in the mathematics as an MOD function which gives the remainder when we carry out a division of one number by another. For example, 13189 MOD 23 is 10, ie 13189/23 is 573 remainder 10.

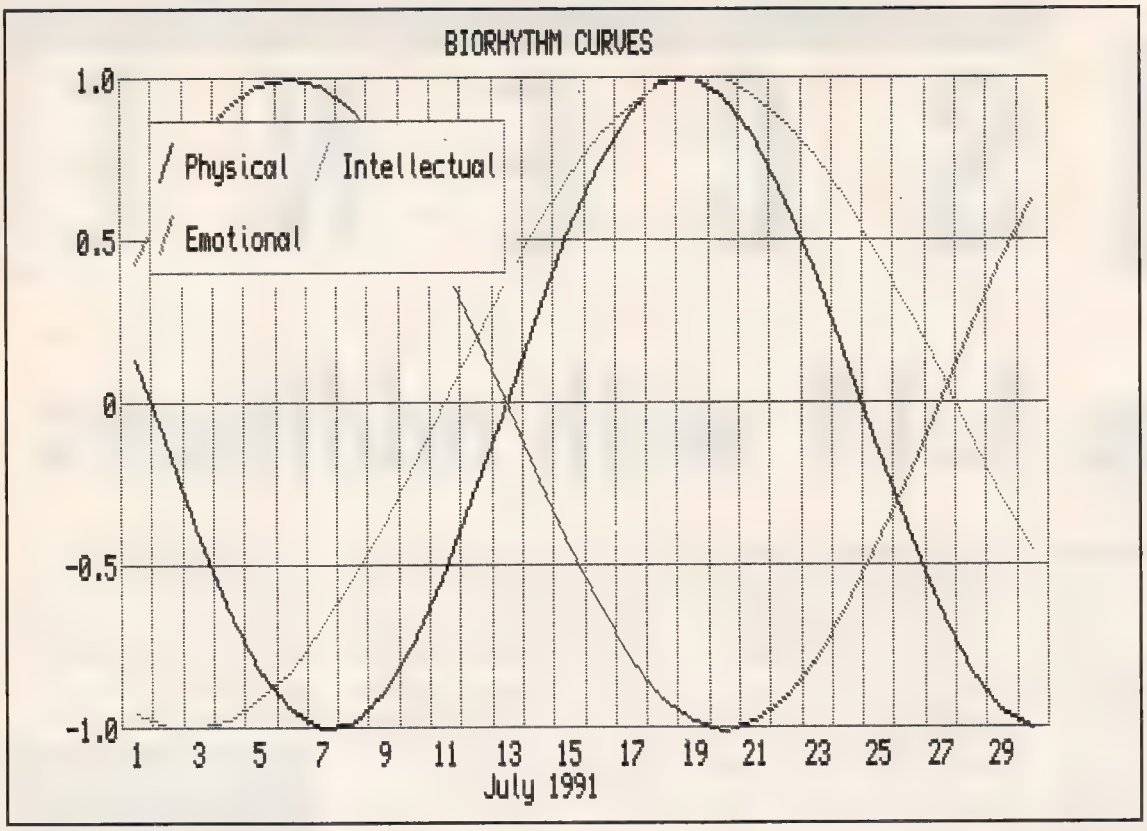
Substitute MOD

Unfortunately the Psion programs do not possess either DIV or MOD functions, but we can construct a substitute with the help of the int( ) function which returns the integer part of a real number. Thus int(13189/23), equivalent to 13189 DIV 23, gives the result 573. The remainder is found by subtracting 573 times the period 23 from 13189:

```
13189 MOD 23
```

is equivalent to

```
13189 - 23*int(13189/23)
```



which gives the result 10. Now we can include this phase shift in our formula to get the correct biorhythm curve on any date:

```
Physical=sin(2*pi( )*(cell + (13189-
23*int(13189/23)))/23)
```

Similarly, the emotional and intellectual curves are given by the formulae:

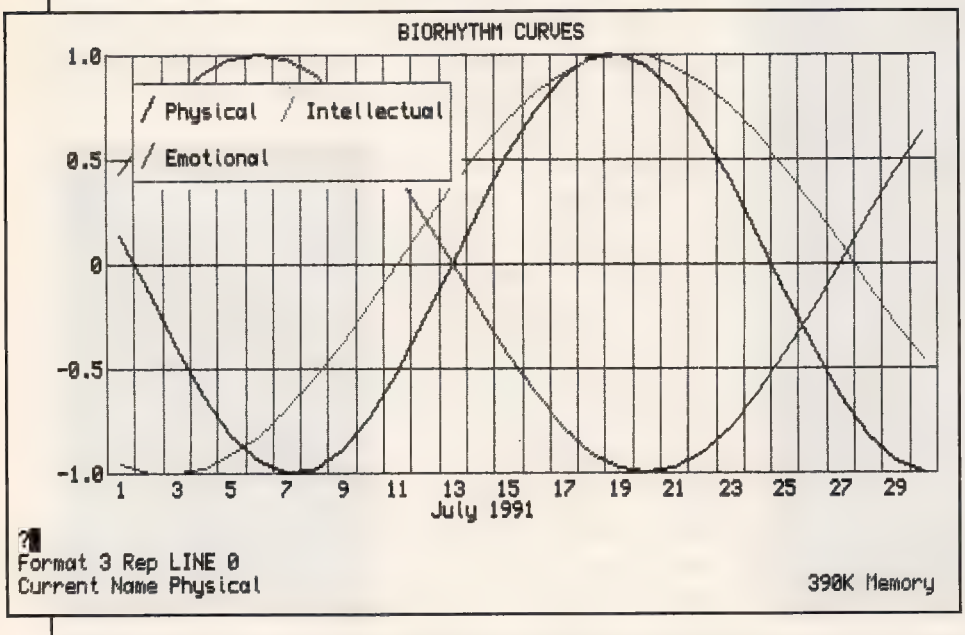
```
Emotional=sin(2*pi()*(cell + (13189-
28*int(13189/28)))/28)
```

and

```
Intellectual=sin(2*pi()*(cell + (13189-
33*int(13189/33)))/33)
```

Barchart

If you type these formulae straight in after loading Easel, you will get a barchart labelled January to December which is both unsatisfactory to the eye and potentially misleading since Easel's built-in month labels have nothing to do with the period to which the curves are meant to refer. Remember that our formulae generate the curves for each day starting with today's date.



To improve the appearance, start by changing some of Easel's default values before typing in the formulae. First, change the format from bar-chart to line graph by pressing F3 C F 3 ENTER to C(hange) to F(ormat) 3 and then press C G 0 ENTER to C(hange) the G(raph) paper back to format 0 (easiest for viewing). Remove the labels by pressing E L to E(dit) the L(abels) and then press F4 twelve times to remove them one by one. This will have the added effect of increasing the number of cells visible when F4 pressed is for the twelfth time. When Easel has finished redrawing the screen, press F4 a further eight times to get 30 cells (days) showing. Alternatively, in data-entry rather than command mode, press TAB to move the cursor across the screen to get the extra cells. Press ESC to return to command mode to edit other items such as axis and graph titles or just press ESC again to return to data-entry mode.

Now enter the formulae given above for the Physical, Emotional and Intellectual biorhythm curves. After entering each one, press F3 C L (number) ENTER ESC to C(hange) the L(ine) colour to whatever number you wish and return to data-entry mode. I find 0 for white, 1 for green, and 2 for red are easy to distinguish. Then press F3 V ENTER ENTER to V(iew) all Figures with format 3. Enter F3 K ENTER to K(ill) Easel's ready-made 'figures' data set.

You can then tidy up the graph further by editing the text and axis-labels and repositioning the key, or removing it if you prefer, as shown in the figure. The instructions for carrying out these operations are described in the QL User Guide. You could make a screendump of the graph to a dot-matrix printer from within Easel by pressing F3 P P. You could update it monthly by typing the formulae again with your new age in days. Also, by increasing the number of cells visible on screen, you could get three months' worth of data on one graph.

If, before now, you've experienced days when things seemed to go right for you and also the occasional 'off' day when everything ended in disaster, you may find the answer in your daily biorhythm chart: it might be worth having a look!

QL SCENE

QTop 1.11 with additions

Cowo Electronic have released version 1.11 of their front end for the QL and Thor, *QTop*. Features listed by Cowo include a completely revised Boot program with simpler user configuration, including F5 startup for the Amiga with the QL Emulator; two new Thor XVI Emulators for the Miracle Gold Card, one for pre-JS roms and one for JS and later roms including Minerva; QJump's RAMPRT ramdisk replaces the CST Ramdrive (which is still included for users of standard 128 K QLs); the latest English version of the QJump extended Pointer Environment, including a special version for Thor XVI users, supporting the Thor mouse; configurable programs menu via Config_Bas; and button and hotkey demo files. Other parts of the program have been updated to make the most of ExeQtor and the Gold Card; plain text error messages are included; QTop-Desk can be started directly via a Button. DIRECTORY and the KEYBOARD menu have been revised and rewritten respectively; programs in MYPROGS can have a program path each; VIEW SCR in the TOOLS menu has a menu for action routines, and there are other upgrades of detail.

The UK price for the new version is £29.50, disk only. Registered QTop users should contact Cowo about updates. **Cowo Electronic, Munsterstrasse 4, CH-6210 Sursee. Tel: (Switzerland) 045 211478.**

Menus from Merz

Jochen Merz has told *QL World* that his program *Menu Extension 3* is now in production, with improved file-select allowing display of multi-column files and subdirectories, and a complete tree. A new feature, Directory-select, allows directories to be constructed exactly as found in Qpac 2 or the File-select menus, and gives direct access to any device, subdirectory and the pre-defined directories. List-select is also new, replacing the old select-from-list and allowing standard arrays to be passed to it.

For all QL-Emulator users with an inbuilt Mega ST clock, the bug which adds an extra day to the year every year has been corrected. You can get a free update by returning your QL BOOT disk with two International Reply Coupons to Jochen, or just put an ADATE -86400 in your BOOT file.

The flp/ram Level 2 eproms for the Trump Card and SuperQ Board now both contain the ATR device which allows direct access to Atari/IBM formatted disks, due to special demand. The new versions have risen in price to £18, but, says Merz, that is a more than fair price for what you get. Please specify Trump Card or SuperQ Board when ordering – the new eproms are different.

Orders and enquiries to Jochen Merz Software, Im stillen Winkel 12, 4100 Duisburg 11, West Germany. Jochen can now accept all major payment cards, including American Express.

New England Group in expansion

The New England Sinclair QL Users Group (NESQLUG) has expanded its membership to 35, and its group software library very considerably by acquisition of the German IFE public domain/user group disks. The IFE list comprises some 90 disks and 1230 programs from users all over the world, in many languages (many PD programs on the IFE list appear in other PD libraries and sources). Hugh Howie writes about setting up a Gold Card across the Great Pond, and the prompt and accurate help he received from Miracle Systems with his queries. Another report also points out that the QL's notorious membrane keyboard is ideal for use in a pottery or other industrial situation – it's not sensitive to dust and soot, and has no fan to suck dirt into the works.

Membership of NESQLUG is $US10, students $US5, overseas (air mail) members $US15. Contact the treasurer and membership secretary, **Sherm Waterman, 40 Eileen Street, Yarmouth Port, MA 02675, USA.**

Euro Fair

Following a1991 with no European Microfair at the Eurovolleycentre, Brussels, 'owing to circumstances', QL club BruQsL are now planning the European Microfair 1992. Registration for firms and clubs is now open. For more information, please write, phone or fax Jaques Tasset, Aarlenstraat 104, B-1040, Brussels, Belgium. Tel. (local) 02/2331222, fax 02/2331220

We are at present having difficulties in exporting copies of Sinclair QL World. So that you do not miss any copies, we invite you to take advantage of the following offer:

Why not buy yourself Knock down subscription to your favourite magazine! (•

Subscribe for one year and you will receive 15 issues for the price of 12.

That's 3 issues Free!!

Subscribe for two years and and get 3 FREE issues AND a discount of £4.00!!

Fill in the coupon below and send it with your remittance to: MSM Subscriptions Department, Lazahold Ltd., P.O. Box 10, Roper Street, Pallion Industrial Estate, Sunderland SR4 6SN. (the first issue of a new subscription to be delivered will be one or two issues after the one you placed your order in.)

Please send me my 12/24 month subscription to Sinclair QL World. at the special discount rate of £_____ I will receive 3 free issues before my subscription becomes due for renewal.

I enclose my cheque/money order for £ _____ made payable to MCPC Ltd or debit my Access/ Visa card. No._____ Expiry date _____

Name _____

Address _____

_____

_____(Please enter postcode to ensure prompt delivery)

Signed. _____ Date. _____ QL 0392

EUROPE £32.90 12 mths REST OF £40.90 12 mths
£61.80 24 mths WORLD £77.80 24 mths

Overseas rates include Airmail service

MIRACLE SYSTEMS

MASTERCARD VISA

QL GOLD CARD

£375 inclusive (£330 export)

• 2 year warranty • 14 day money back guarantee •

- The **GOLD CARD** is the ultimate expansion for the QL. It speeds up program execution by a factor of about four, increases the memory to 1920K bytes and enables the QL to interface to 3.2M Byte disk drives (as well as 720K and 1.44M).
- There is a battery backed clock so that the QL powers up with the correct time and date.
- We offer part-exchange for old expansions:

Part Ex. Item	GOLD CARD price	
TRUMP CARD 2	£225.00	(£200.00 export)
TRUMP CARD 768K	£275.00	(£241.00 export)
TRUMP CARD 256K	£300.00	(£263.00 export)
EXPANDERAM	£325.00	(£284.00 export)
DISK CARD	£325.00	(£284.00 export)
RAM + DISK I/FACE	£325.00	(£284.00 export)
Other Expansion	£350.00	(£305.00 export)

- All the functions on the TRUMP CARD, i.e. TKII, printer buffer, screen dump, RAM disk and memory cut, are carried over to the **GOLD CARD**.
- The floppy and RAM disk drivers now support sub-directories.

JOIN QUANTA !

QUANTA is the independent QL user's club. If you have not already joined then consider the following:

- QUANTA stands for "QL Users and Tinkerers Association" and provides a forum for users to get together.
- Members requiring guidance on software and hardware can seek out expertise from fellow members.
- Local sub-group meetings are regularly held throughout much of Britain.
- There are also workshops where we and others demonstrate our wares. Members have access to a large software library and can often find spare parts and second hand bargains at QUANTA meetings and can see new products.

The £14 (£17 overseas) subscription includes an informative monthly magazine. Most of all, though, it is good fun so contact the Membership Secretary, Bill Newell, 213 Manor Road, Benfleet, Essex, SS7 4JD, or telephone (0268) 754407 now!

Tel: (0904) 423986

To place an order by phone please have your credit card ready. For overseas customers we charge the prices shown in brackets.

To order by post, please fill in the form opposite or write to us quoting your credit card number and expiry date, or enclosing a cheque payable to MIRACLE SYSTEMS Ltd.

To: MIRACLE SYSTEMS Ltd, 25 Broughton Way, Osbaldwick, York, YO1 3BG, U.K. Tel: (0904) 423986

Please send me _____

I enclose a cheque to the value of: £ _____

Or debit credit card ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

expires ☐☐/☐☐

Name _____ Signature _____

Address _____

_____

_____

MIRACLE SYSTEMS

MASTERCARD VISA

EXPANDERAM 512K

£50 inclusive (£46 export)
1 year warranty

This simple expansion offers a great improvement to users of unexpanded QLs. It increases the QL RAM from 128K to 640K. The effect of this on Quill, for example, is threefold:

- Much larger documents can be created
- A useful increase in the speed of operation
- The need for def_temp files is removed

Also the QL expansion connector is replicated so that a disc interface can still be connected. n.b. The EXPANDERAM cannot be used with the TRUMP CARD or GOLD CARD

3.2 MBYTE DUAL DRIVE

If you currently have an order with us or are considering ordering the 3.2 M byte DUAL DISK DRIVE then please note that we are experiencing difficulty in obtaining the mechanisms at a suitable price. This has led to long delays in delivery of this product. We sincerely hope that you will agree with us that we should not pay over the odds for goods imported from Japan and that you will bear with us until the negotiations can be resolved. Please also note that when used with the TRUMP CARD diskettes will only give capacities of up to 720K bytes.

QL CENTRONICS

£29 Inclusive (£28 export)

The QL CENTRONICS PRINTER INTERFACE enables a printer with a standard Centronics parallel port to be connected to the QL with great ease. the interface plugs into the Centronics port on the printer and has a 3 metre cable terminated by a QL serial plug. The serial plug can be plugged into either SER1 or SER2. Nothing else is required.

We are exhibiting in:

GLASGOW:

8th March 1992
City Hall
Candleriggs

LONDON:

14th March 1992
Royal Horticultural Hall
Greycoat Street,
Victoria

GERMANY:

21st March 1992
Münster

MIRACLE SYSTEMS Ltd, 25 Broughton Way, Osbaldwick, York, YO1 3BG, U.K. Tel: (0904) 423986

SOFTWARE FILE

PCB_CAD

INFORMATION
Program: *PCB_CAD 2.01*
Supplier: *Lear Data Systems*
Price: *£69.95*

Ray Dawson has been using Lear Data Systems' package to create printed circuit board designs.

PCB_CAD is a program specifically designed to create two-dimensional drawings for use in the making of printed circuit boards. It can also be used as a general purpose drafting package, and I have used this facility to prepare the component layout drawings for most of the pcbs I have designed and made.

On receipt, the disk contains eighteen files. One of these is the instruction manual, in the form of a *Quill* document file. This is logically formatted, ranging from the contents page at the front, to two appendix sections at the end, and is easy to follow. It contains all the information necessary to operate the program successfully. Additionally, the appendix contains a list of all the components available in the on-line library file, as well as details of the art file data format. The art file is the file that you create and, when saved to disk, it is automatically given the extension _art. The size of this file can be displayed at any time.

The instruction manual rightly advises that, as this file contains your precious art work, you should keep backing it up at regular intervals as you proceed. The other files on the disk are the Boot and Backup programs, a configuration program, various printer drivers including Shinwa CPT-80, Panasonic KX-P1124, Canon PW1080A and PW-1156A and the Epson LX-800, a public domain PostScript conversion program supplied free of charge, along with a toolkit extensions program required for the correct operation of the suite. In addition, I understand that later versions of the program also contain conversion programs for Gerber and Wanatabe formats. The on-line library and Help files, and finally a couple of demonstration _art files in the form of a pcb design and a circuit diagram complete the list.

Artwork is created by manipulating the elements of the drawing – tracks, pads, blocks or text – using the cursor keys or, as in my case, a Mouse. As one of the 16 overlaid layers can be worked on at any time, each layer is allocated a different colour – seven of these being active at any one time. The colour of each layer may be altered using the Mask command and, as the background colour is black, any layer may be hidden from view by changing it to black. However, when this is done, it disables the Find command. This command enables any element to be 'picked up' and edited. As this is done, the element changes colour to white and cyan, defining the start and finish, or head and tail of the element.

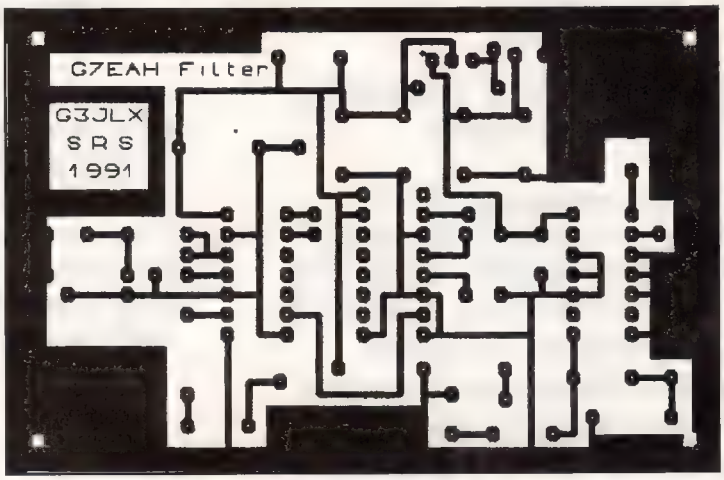


Figure one: a pcb layout as produced on PCB_CAD (track side).

The pcb diagrams are printed here slightly larger than life-size. PCB_CAD originates them to full technical tolerances.

Library elements

If the element happens to be a library item, its library identity number can also be established at this stage using the Identity command. Until I acquired a colour monitor, I found the program very difficult to use when trying to identify more than one overlaid layer, or the colour change when editing. This is one of the very few criticisms I can level at a very professional program, but I must emphasise that a colour monitor is essential if you are contemplating any really serious multi-layer work. As an area at the bottom of the screen is already used to display such things as the current cursor co-ordinates and the grid step setting, I feel that it would be helpful to see the current active layer number displayed as well, to assist users with monochrome monitors.

Completed or unfinished art work can be stored on disk for later processing, using the Save and Load options, while the library file provides a source of the more commonly used elements, such as dual inline IC bases. This file can be extended to contain up to a maximum of 32000 items. The latest version of the program, I am told, has simplified this considerably. In version 2.01 it is necessary to exit the program in order to add to the library, but I understand that later versions permit additions to be made without re-booting.

Output can be either as HP-GL (plotter) files for disk storage or serial output, or as a PostScript file for output to a laser printer. In addition, as previously mentioned, later versions of the program contain conversion programs to Gerber and Wanatabe format so that all possible graphics reproduction devices are now catered for.

With tongue somewhat in cheek, I have to admit that I have not yet fully explored the whole drawing area of 32 by 32 inches, or the claimed resolution of 0.001 inch, but I can verify that the resolution is well within the limits needed to produce very acceptable quality boards, even when using the latest surface-mount technology.

The hardware requirements are either a QL or Thor with a minimum 512K expansion memory, a monochrome or colour monitor (bearing in mind my previous comments),

and an Epson compatible dot-matrix printer, flat bed plotter, photoplotter or laser printer. I use a Brother M1009 printer, and have got perfectly acceptable results using the Epson LX-800 printer driver supplied, and the methods explained later.

Manual advice

The manual advises that 'the best way to learn is to sit down in front of the screen and try out the commands'. This is very good advice indeed, and it is by this method that I have become acceptably proficient in its use. At any time the F1 key will always display the list of commands available at that point in the program, and I have found this on-line help invaluable. At switch-on it is immediately evident that, although the program is not copy protected, some attempt has been made to personalise your copy by the inclusion of your name at the top of the start-up screen – a nice touch, and one which engenders a sense of ownership while at the same time ensuring that the copy is unique to the purchaser.

Initially, the Normal Edit Mode is in operation, displaying a cursor in the form of a cross. The x,y co-ordinates of this are displayed at the bottom of the screen – at 0,0 to start with – and these are dynamically updated as the cursor is moved about the screen, the distance moved with each key press depending on the selected Grid increment setting. At any time, the cursor may be positioned centrally on the screen by use of the space bar, or the current cursor position redefined as the 0,0 co-ordinates by invoking the Relocate command. The Grid increment setting is adjustable in eight pre-programmed steps, ranging from one thousandth of an inch to one inch, the intermediate steps being at 5,10, 20, 25, 100 and 200 thou. A grid marker in the form of a trail of blue dots may be toggled on and off with the F3 key at any time.

Tracks and pads

Tracks, pads, blocks and text can be manipulated in size and, in the case of the pads, shape as well – either square or round – can be selected. The sizes range from 50 thou. to 260 thou. in sixteen intermediate steps. The inclusion of a 'bomb-sight' board maker, in the form of a cross within a 200 thou. diameter circle, completes the pad library. Remember that each element has start and finish positions designated as the head and tail. During editing, these designations may be exchanged, or the cursor moved from one end to the other by the use of a single key stroke, removing the need for a lot of cursor manipulation. The Kill command removes the element completely, while Delete removes just that part of the element to which the cursor is currently attached, leaving the rest unchanged.

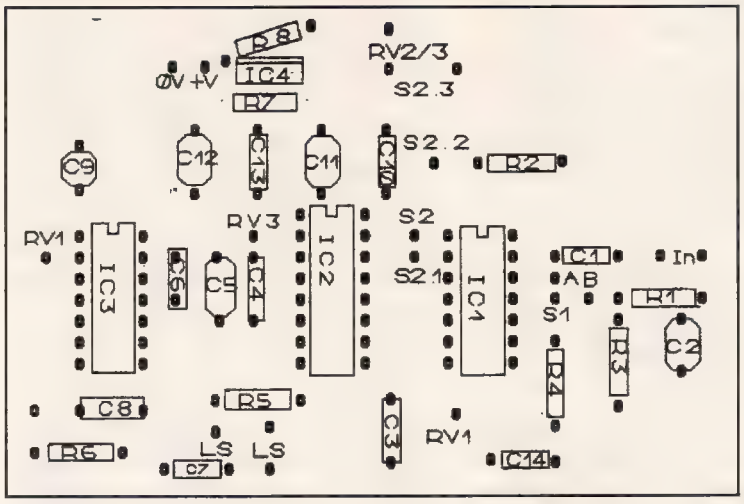


Figure two: How PCB_CAD lays out components for use on the board.

Track thicknesses can be from varied, in 5 thou. increments from a minimum of 5 thou. up to 315 thou. These are designated Track 1 to Track 63. In addition there is a Track 0 with a width of 2 thou., and I found this especially useful in the preparation of circuit diagrams and general drawings.

Blocks are areas of the board defined by the use of the cursor keys after selecting the Block edit mode. These areas represent solid blocks of copper on the finished board, and are invaluable for drawing earth-planes around other circuit areas.

Text is also fully programmable in size and shape and is a very attractive 'draughtsman'-style typeface. Nearly all operations are performed by a single key stroke (sometimes with the addition of the Ctrl key) and this makes a very user friendly environment. The initial letter of its name is usually sufficient to execute the desired command, but you will have to get your own copy of the program to find out how conflicting words like Library and Layer have been dealt with.

If library components are to be edited, there is a Library Unlock command to enable this to be done, otherwise a library item, even though it is individual elements may be physically separate, will behave as a unit. In addition to editing individual elements, Global edits may also be performed, manipulating whole areas. Because this is a very powerful command, an Undo command has been included though care is still needed – so be warned. It is worthy of note at this point that track widths and also pad sizes may be globally changed, but care is needed here, as some changes may be irreversable by global means, and may leave you having to do individual element edits to achieve the desired result. Facilities to copy (or duplicate) areas are especially useful when preparing drawings of memory planes or similar repetitive designs.

Rubber-banding

In the Element edit mode components brought from the library, or individually created, may have their terminals interconnected using tracks rubber-banded from point to point. I have found that it is best to use right-angled tracks wherever possible, as the use of the sloping tracks does tend to produce jagged lines on my dot-matrix printer. However, if you possess a plotter or other more advanced means of reproduction, then no such restriction need apply.

Individual elements can be modified, if

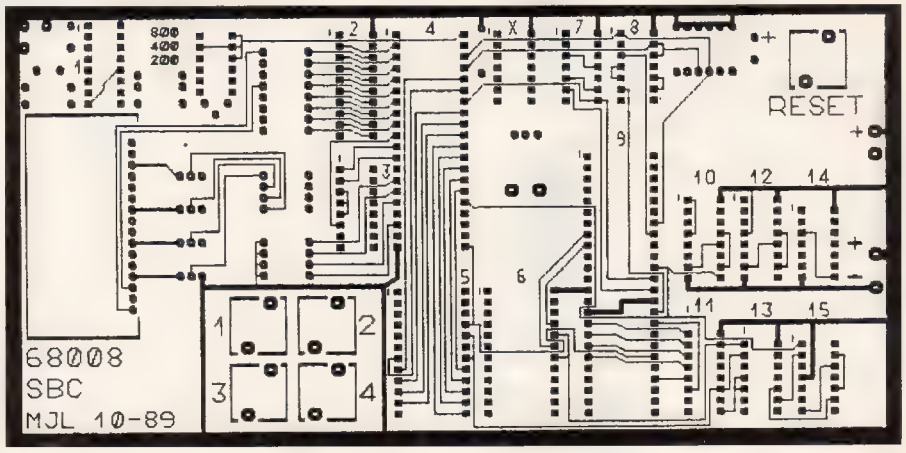


Figure three: One side of a double-sided pcb from the sample file.

desired, or the element moved around the drawing area as an entity. Alternatively, a whole area can be selected for modification or positioning, by entering the Area formatting mode. Rotation of the whole, or part, can be made clockwise or anti-clockwise, or on either X or Y axis. The axis or origin of the rotation depends on the cursor position when the command is executed.

In preparing a drawing I have found it best to design the whole layout on graph paper first. If a graticule spacing of one tenth of an inch is used, it is then very easy to translate this to the screen, with a grid setting of 100, as nearly all electronic components can be made to fit on a one-tenth inch matrix. Have the components to hand and measure them as you go. When the design is complete, sit down and carefully transfer it to the screen. On completion, do a draft print. Look carefully at the joints between tracks and pads. By experience I have found that it is best to use the highest possible scale magnification when joining tracks to pads, to ensure that they connect, but do not extend into the centre region of the pad.

Scale factors

Scale factors are controlled by the numeric keys 1 to 8 for the selection of a particular magnification. Alternatively, the plus and minus keys allow single-step magnification changes up or down. Should the track extend too far into the pad it will print as a solid block. Leaving a trace of white in the centre of the pad helps to locate the drill when it comes to drilling the etched board later in the process.

By way of example, Dr. Paul Stewart G7EAH recently published a design for an audio notch filter. Amongst others, I have produced a printed circuit board based on this design, using PCB_CAD, and copies of the board have been used successfully by members of our local amateur radio society in Stockport. The full size print of the board design is shown in **figure one**.

To make the printed circuit board, the art-work was first printed using the mirror image, that is, first rotated through either the X or Y axis and printed using a times two scale factor selected from the printer-driver options screen. Various parameters are available for the choice of output. By selecting a magnification of two and also a graphics density of two, the printer produces an image with each line double printed, which, with a fairly new ribbon, produces a reasonably dense image.

I have found that two passes is the optimum. Increasing this only introduces smudging, and a single pass is a bit too grey to get the required contrast. Always use a good quality copying paper of about 80gm/m^2 with a matt finish, as this tends to absorb the ink without bleeding. When duplicated on a copying machine at the local office supplies shop on to acetate film at a 50% reduction, this produces a positive image when viewed *through* the film. This not only produces a very accurately dimensioned image, but also has the effect of doubling the density and so far has proved highly successful on every board I have made. Incidentally, the idea of duplicating the mirror image is to get the ink on to the side of the acetate which is going to be in contact with the photo-sensitised surface of the printed circuit board, when it is exposed in the UV light box. This avoids any possibility of the light rays diffusing when they pass through the acetate before reaching the board – think about it, and you'll see what I mean. This program, when used in any of the normally accepted ways, is capable of producing pcbs which are not only dimensionally correct but of the very highest quality. **Figure two** illustrates the way the program was used to help in the placing of components on the finished pcb. PCB_CAD does not do automatic routing, neither does any other program I have seen at the price, but provided one is able and willing to do the basic work, it is in the highest traditions of programming. Much thought and effort has gone into its design with the user in mind.

The demonstration designs included on the program disk speak for themselves. In the interests of space these illustrations have been somewhat reduced in size. **Figure three** shows just one side of a double-sided board design from the Example_art file supplied with the program. **Figure four** shows the component layout for this design. The whole file consists of four overlaid layers. Layer 4 has been used for the top side of the board with all it is interconnections and necessary text; Layer 2 has been used for the component layout diagram; Layer 1 contains the bottom side connections, while Layer 5 has been used for all pads and IC connections, and is printed in combination with Layer 4 and Layer 1 to produce the completed designs for the top and bottom sides respectively.

Dodge duplication

As the pads and IC connections on Layer 5 are common to both sides of the board, a lot of unnecessary duplication has been avoided by using this method. These layers have to be printed out carefully, either independently or in combination, and either 'as seen' or in mirror image in order to achieve the desired end result. This is one of the main strengths of the program and enables absolute registration to be maintained between layers at all times including 'vias' from one layer to another. It will be seen that the sloping tracks have not reproduced very well with my printer, but with a board of this complexity a trip to the local reprography shop or agency would be worthwhile for them to produce the finished art-work using your HP-GL (or other) file on more sophisticated equipment.

I feel that with this program, Lear Data Systems have established themselves as a major contributor to the QL software scene and are to be congratulated on their achievement. Not only does the program perform magnificently, but the price is also very realistic considering its complexity.

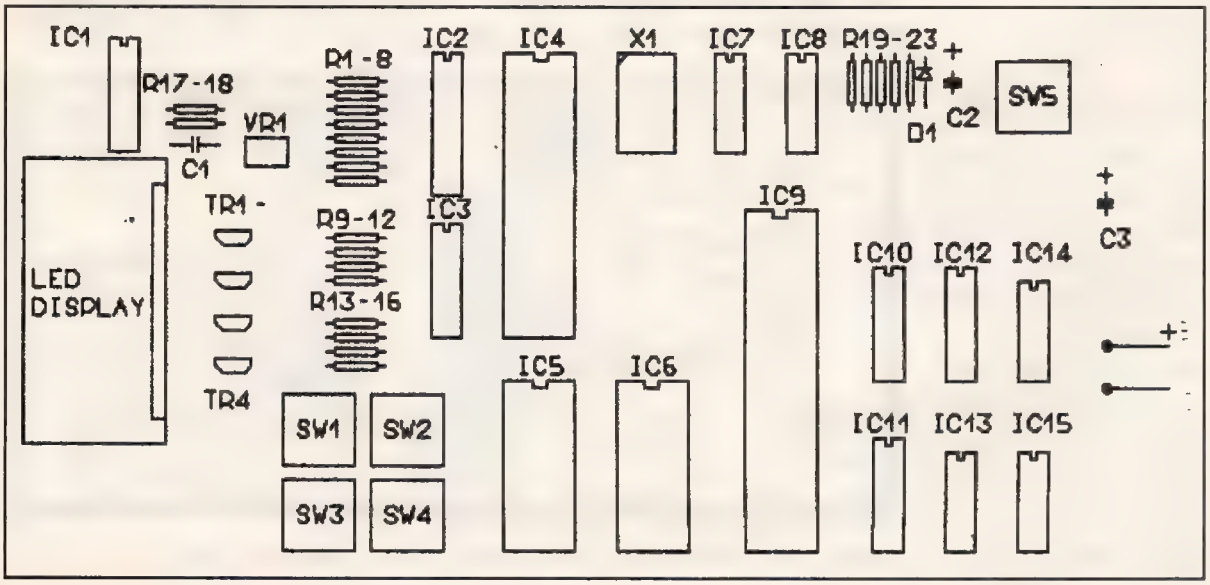


Figure four: The components laid out for the sample board.

SINCLAIR QL WORLD

THE NEW USER GUIDE

SECTION THIRTEEN

KEYWORD INDEX

This month in the Keyword Index, Mike Lloyd moves from DATASPACE to DEVICE_STATUS. Not photogenic, but fascinating.

DATASPACE (file$)
[TURBO TOOLKIT]
DATA_AREA kbytes

MEMORY FUNCTION

file$ — a valid file name

kbytes — an integer constant (not a variable)

When a SuperBasic program is compiled with *Supercharge* or *Turbo*, a set amount of memory is allocated for its personal data. The amount of dataspace can be adjusted from the compiler's control panel and should be made sufficient to hold all variables, arrays and scratch space without wasting any memory. Alternatively, the compiled task can contain a DATA_AREA command which changes the control panel setting. The DATA_AREA parameter represents whole numbers of kilobytes, so DATA_AREA 6 will allocate 6K of memory.

The DATASPACE function returns the number of bytes of memory a task is currently allocated. It returns standard error codes if the filename it uses is not found. Suppose a compiled program which needs 4K for its own variables is about to be loaded to handle a large, user-defined, numeric array. DATASPACE can be used to warn of insufficient dataspace in the following manner:

```
100 DEFine FuNction check dataspace (rows, columns, program$)
110 LOCal array_bytes
120 array_bytes = rows * columns * 7 + columns * 2
130 IF array_bytes + 4000 > DATASPACE (program$)
140    PRINT "Array is too large for dataspace"
150    RETurn 0: REMark "false""
160  ELSE
170    RETurn 1: REMark "true"
180 ENDIF
190 END DEFine
```

DATE
DATE$ (number)
DAY$ (number)
[Superbasic]
DATE (yr,mon,day, hr,min,sec)
[Minerva]

INTERNAL CLOCK FUNCTIONS

number — a large, whole number

yr — an integer year (eg 1961)

mon — an integer month in the range 1-12

day — an integer day in the range 1-31

hr — an integer hour in the range 0-23

min — an integer minute in the range 0-59

sec — an integer second in the range 0-59

For a computer unable to remember which day of the week it is from one working session to the next, the QL has quite a powerful set of time-related commands. The basic premise is that the QL's clock is counting the seconds between 1 January 1961 and 27 February 2097. This is fine for the average business application, but can be restrictive for historical and astronomical applications. Because

numbers such as 9632736 are not immediately recognisable as '7 July 1992' SuperBasic includes a number of functions which translate between the internal representation of time and more normal conventions.

In SuperBasic, DATE is a function without parameters (although it may optionally take brackets to please the purists) which returns the QL's date in its internal, numeric representation, so that PRINT DATE might return 9.632736E8. QView, who developed Minerva, have added an additional facility so that DATE can take six parameters representing year, month, date, hour, minutes and seconds respectively, converting a known date and time into the QL's internal value.

To see the same date in a more understandable format the DATE$ function can be used. DATE$ will convert a numeric parameter into a date and time string such as '1961 Jan 01 09:00:05'. If no parameter is provided, the QL's internal date/time counter is converted into a string.

Programmers sometimes attempt, incorrectly, to slice the date/time string, forgetting that DATE$ is a function which expects a numeric parameter and is therefore thoroughly confused by instructions such as PRINT DATE$(6 TO 8). To extract a part of the string it must first be assigned to a variable. The following function shows how the month can be extracted from a date. The function takes a numeric parameter and returns a string.

```
100 DEFine FuNction GetMonth$ (dateval)
110 LOCal datetext$
120 datetext$ = DATE$ (dateval)
130 RETurn datetext$ (6 TO 8)
140 END DEFine
```

SuperBasic helps with the day of the week, which would otherwise be difficult to calculate, by providing a DAY$ function. DAY$ returns the day of the week represented by the QL's internal clock setting (if no parameter is given) or the day represented by a numeric parameter

DDOWN dirname

[Super Toolkit II]

Directory Navigation

dirname — a valid filename extension

Super Toolkit II implements a multiple-level directory system which was dormant in the original release of Qdos. DDOWN is a command to move 'down' a subdirectory by appending dirname to an existing directory default. It only affects the prefix set by DATA_USE unless PROG_USE has been set to exactly the same prefix, thus 'binding' both the program and data directory pointers. The following commands and their associated remarks should clarify DDOWN's effect:

```
100 DEST_USE scr: DATA_USE flp 1_
110 COPY filename: REMark flp1_filename copied to screen
120 DDOWN work
130 COPY draft: REMark flp1_work_draft copied to screen
```

DEALLOCATE basebyte

[Turbo Toolkit]

Memory Command

basebyte — A valid, even memory address

Areas of the QL's common heap of unallocated memory can be reserved with functions such as RESPR which return the address of the first byte of memory to be reserved. Hold on to this value, because without it memory cannot be deallocated. Turbo Toolkit includes the command DEALLOCATE, which must be followed by the address of the first byte, to return a reserved memory area back to the common heap. Avoid memory fragmentation, where allocated and free chunks of memory are interspersed with each other, by releasing memory in the opposite order in which it was allocated.

DEFAULT_DEVICE

drive
[Turbo Toolkit]

Device Command

drive — A drive prefix with final underscore

Turbo Toolkit does not acknowledge the Qdos directory hierarchy but instead allows programmers to declare a default device. This means that a command such as OPEN filename can be issued without error, provided that a command such as DEFAULT_DEVICE flp1_ has been issued earlier.

DEFine FuNction

fname (param1, param2, ...)

DEFine PROCedure

pname (param, param2, ...)
[User-Defined Keywords]

fname/pname — A unique, non-reserved work

param — (optional) Parameter names

When SuperBasic was first introduced the DEFine FuNction and DEFine PROCedure constructs justified to a large degree the 'Super' prefix. Most of SuperBasic's predecessors, and even some of its successors, did not permit users to extend the language in any meaningful way. Where other Basic dialects, such as BBC Basic, allowed procedures to be defined they were not as complete as SuperBasic.

A user-defined procedure or function can do anything which can be defined in terms of existing SuperBasic keywords. Once defined, it can be used by other user-defined structures. If a program has a need for text to be centred in a window, SuperBasic has no command of its own to perform the task, but it does allow programmers to develop one:

```
100 DEFine PROCedure CPRINT (text$, windowwidth)
110 REMark windowwidth gives window width in characters
130 IF LEN(text$) >= windowwidth
140   PRINT text$
150 ELSE
180   PRINT TO (windowwidth-LEN(text$))/2; text$
190 ENDIF
200 END DEFine
```

Functions differ from procedures in that they return a value of some sort and that their names must agree with the type of value returned: the name of a function which returns a string must end in a $.

SuperBasic does not include a function which returns the sign of a value, but a user-defined function can be written which does so:

```
200 DEFine FuNction SIGNUM (value)
210 IF value > 0: RETurn  1
220 IF value < 0: RETurn -1
230 RETurn 0
240 END DEFine
```

Functions and procedures in SuperBasic are fully recursive in that they can call themselves either directly or via a subordinate procedure or function. Although clever and concise code can often be written in this way it is usually preferable to avoid recursion because of the overheads it imposes on the interpreter and the ease with which undesired results can be obtained. However, to show an easy recursive solution to a problem, here is a procedure to print out each word of a long string on a separate line:

```
300 DEFine PROCedure Wordlist (text$)
310 split = " " INSTR text$
320 IF split = 0
330   PRINT text$
340   RETURN
350 ELSE
360   PRINT text$ (1 TO split)
370   Wordlist text$ (split+1 TO)
380 ENDIF
390 END DEFine
```

The basic structure for a user-defined procedure or function is an opening line declaring the name of the structure and any parameters it may have. Definitions do not need to have parameters if they are not wanted. Parameter lists are separated by commas. Note that both procedures and functions require brackets when the definition is called. There then follows a body of lines which define the actions carried out by the user-defined structure. The final line is END DEFine. SuperBasic allows the body of a procedure or function to contain another user-defined segment, but it would be loose programming practice to take advantage of it. Similarly, SuperBasic does not object if a GOTO or GOSUB within a definition jumps to some code outside the definition boundaries, but it would be an incautious programmer who thought that this was a worthwhile activity.

An unusual feature of user-defined parameters is that their names do not have to conform to the normal SuperBasic naming rules. Thus, although the above example uses text$ for its parameter, text would have done equally well. Unless a parameter can represent either a string or a number it is best to stick to the normal naming conventions.

The name given to a definition becomes in effect a new SuperBasic keyword, and can be used as such, as shown in the following examples:

```
500 CPRINT "hello world", 32
550 X = Y * SIGNUM(Z)
600 PRINT userfunction (15, 20)
```

DEG(radians)

[SuperBasic]

Trigonometry Function

radians — A numeric value

In line with most computers, the QL is happier dealing with radians than with degrees. However, people are usually more familiar with degrees and so conversions between the two are part of the QL's trigonometry suite. DEG converts an angle expressed in radians into its equivalent in degrees.

The ratio between the radius of a circle and its circumference is slightly more than 6:1. In other words, six and a bit lines each the length of a radius would be needed to cover a circle's circumference. This ratio is better expressed as 2 *pi. It should be obvious that DEG (pi) will return 180 and DEG(2 * pi) will produce 360.

DELETE filename

[SuperBasic]

FILE COMMAND

filename — A valid filename

DELETE is a straightforward command used to remove a file from a directory. Provided that it is followed by a technically valid filename it will issue no warnings or error messages, even if the file does not exist. QL users usually find out to their regret that computers have absolutely no idea about the value of a file, and will blindly delete a large, unique and essential file with as little fuss as a small, inconsequential one.

Fortunately, the DELETE keyword only affects the reference to the file in the disk or microdrive directory, which allows a file repair utility to recover a deleted file provided that its contents have not been overwritten by subsequent saves. A couple of years ago a drug dealer was successfully prosecuted and imprisoned because he believed that DELETE removed all evidence of his database: the police recreated the directory as it was before the delete was carried out and recovered all the incriminating information from the computer.

DEL_DEFB
[Superb Toolkit II]

MEMORY MANAGEMENT

DEL_DEFB takes no parameters. It provides an effective cure for a form of common heap fragmentation. The QL copies directory contents into the common heap area of its memory in order to reduce the access time to reach files. Unfortunately, this habit mixed with user-sponsored memory allocations can fragment the memory so that large chunks of it become temporarily unusable. DEL_DEFB removes all file definition blocks from the common heap. It should not be used when channels are opened to any disk or microdrive.

DESTD$
DEST_USE device
[Super Tookit II]

DEVICE MANAGEMENT FUNCTION

device — A valid Qdos device or file name

The DESTD$ function reports which device or directory is currently the default destination device. In most cases, the default will be the screen, but it can be changed to the printer, network or a file. The destination device is set with the DEST_USE command.

DEVICE_SPACE (chan)
[Turbo Toolkit]

DEVICE MANAGEMENT FUNCTION

chan — An open channel (leading # is optional)

Before sending data to a file it is convenient to note beforehand whether there is likely to be enough space available to hold it. The size to which a file can grow is determined by the room left on the device which holds it. DEVICE_SPACE assumes that you have opened a channel to a file on a disk or microdrive. It returns the number of bytes left on that device. If the channel refers to a non-storage device such as the network, the screen or the printer DEVICE_STATUS returns a very large number. See DEVICE_STATUS below for a more useful method of obtaining this information.

DEVICE_STATUS
(type,file)
[Turbo Toolkit]

DEVICE MANAGEMENT FUNCTION

type (optional) — an integer representing access type
file — a valid filename

A major limitation to SuperBasic is its lack of error-recovery facilities. All of the QL roms had incomplete WHEN ERROR code which was subsequently completed by Super Toolkit II and by the Minerva roms. However, Digital Precision's compilers needed to have version-independent error-trapping code. Most recoverable errors relate to file access, and so the DEVICE_STATUS function was created. You can provide the filename and, optionally, the sort of access you require to the function and its code will test the device to determine if your intentions can be fulfilled. Any positive return value indicates that you can go ahead, and incidentally reveals the amount of available space on the device. A negative return value can be interpreted as a standard error code.

The access codes are as follows:

0 OPEN
1 OPEN_IN
2 OPEN_NEW
-1 OPEN or OPEN_NEW

The possible error values returned by DEVICE_STATUS are:

-3 Insufficient memory
-6 Insufficient memory
-7 No such device or file
-9 Device busy
-11 Device full
-12 Device is valid, but the filename is illegal
-16 The infamous 'bad or changed medium'
-20 Device is write-protected or being read

PCB CAD

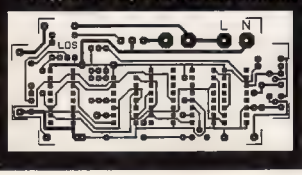
FROM

Lear DATA SYSTEMS

A FAST AND EASY TO USE PCB DESIGN SYSTEM FOR THE QL AND THOR

DESIGN FEATURES INCLUDE:

1. A 32 by 32 inch drawing area.
2. Resolution down to .001 of an inch.
3. Up to 16 separate layers with variable colour assignments.
4. 16 predefined Pad shapes and sizes.
5. 64 Track widths from .005 to .32 of an inch.
6. The number of components on a layout is limited by memory only.
7. Auto repeat on all elements - ideal for memory planes.
8. Can be used for surface mounted devices.
9. Auto via system for multilayer boards.
10. Over 32000 user definable library components.
11. Can output plot files to any valid QL/THOR device.
12. Produces Dot matrix, HP-GL or laser files.
13. Supports Trump card and ABC Electronik keyboard interface.



Minimum system requirements: 512K expansion and 1 disc drive.

Only **£69.95** PRICE INCLUDES P/P

Please make cheques payable to Lear Data Systems and send to:

Lear DATA SYSTEMS

6 SOUTH VIEW GREEN, BENTLEY, IPSWICH, SUFFOLK IP9 2DR.
TELEPHONE: (0473) 310804

☆ sinclair QL ☆

EEC LTD THE WORLD'S LARGEST QL RETAILER

The expandable system for businesses, beginners and experts.
COMPLETE QL's - JS ROM - TESTED PRIOR TO DELIVERY - 6 MONTHS GUARANTEE
Package includes PSU, TV Lead, QL Software 2.35: Quill - Wordprocessor, Abacus - Spreadsheet Archive - Database & Easel - Business Graphics, Comprehensive instruction manual.
One year's free membership to QUANTA - the independent user group who provide help, newsletters & a library of 400+ programs - mostly free.

£125

BARGAIN PRICE MONITORS

All prices include 17.5% VAT

To keep your costs down, the price you see is the price you pay -
★ *WE HAVE BEEN SEARCHING FOR A HIGH QUALITY QL MONITOR FOR A LONG TIME, WE THINK THE GOLDSTAR IS THE ONE!* ★

OFFER 1 - colour
GOLDSTAR 14" Hi-Res with 0.31 dot pitch in stylish grey non glare screen (also O.K. for PCs) (Tilt & Swivel base £5) £199

OFFER 2 - colour
Philips 8833 Mk II - a lower resolution monitor than the GOLDSTAR but with stereo sound capabilities. 0.42 Dot Pitch. £279

OFFER 3 - Mono-DIGIVISION with QL Connector
12" Green Screen monitor, 2 Colour case £49

QL TO PC KEYBOARD ENHANCEMENTS

QL/PC 102 keyboards - £25 UK version, - £35
QL/PC keyboard interface - £75 External fitting box & cable - £17

PRINTERS *Centronics - Serial Interface £27 if reqd.*

All prices include 17.5% VAT

Olivetti/TA dot matrix, up to 200cps/30 NLQ, FX850 emulation - £129
Canon BJ10ex bubblejet, very high quality (360 dpi), A4 size - £259
OKI microline, ML183, wide carriage dot matrix printer - £129
We can supply any make/model printer - please ask for details.

QL/Spectrum spares & repairs - please ask for details

EEC LTD

18-21 MISBOURNE HOUSE, CHILTERN HILL, CHALFONT ST PETER, BUCKS. SL9 9UE

Tel: 0753 888866

Payment terms:
CWO, Access, VISA et cetera
Cheques - allow 10 days
Delivery:
Carriage - £9 Postage - £3
Product subject to availability. E&OE.

Fax: 0753 887149

QUBBESoft P/D

Public Domain and Shareware Software For The

Sinclair Q.L.

S.A.E. (A5) for Software Catalogue

Copying Fee 50p

Copying Fee 30p

C68 Public Domain 'C' Compiler

Consists of 8 3.5in Disks, containing the following:
Disk1: Main System Disk- Contains the various passes of the Compiler.
Disk2:Utilities-Contains all Utilities associated with C68.
Disk3: Documentation _All in Quill'_doc' format.
Disks 4-8: Source 1 to 5- contains Source Code for C68 Compiler, CC, CPP, R568, LD, Generic part of Standard 'C' Library, QDOS Specific part of 'C' Library. Additional Libraries & Utilities such as MAKE & SLB.

Copying fee: £2.50p (User Supplies Disks) or £5.00p (Pre-Copied Disks)

MicroEMACS V3.9p Text Editor

Consists of 2 3.5in Disks, containing the following:
Disk1: Main Program Disk - Contains EMACS Main prog EMACS_RC Resource file, EMAC5_HLP Help file, EMACS_MSS Userguide, EMACS_TUT Tutorial. EMAC51 _doc to EMACSF _doc user guide in _doc form.
Disk2: Source Code - Contains all relevant Source Code for Micro-EMAC5 V3.9p. Also includes document on the latest update, and how it has been re-compiled using C68.

Copying fee: £1.00p (User Supplies Disks) or £2.00p (Pre-Copied Disks)

Disk SPECIAL 3 & SPECIAL 4

SPECIAL 3: XLisp V1.4a

TPD program has been ported from the IBM PC with only small changes necessary to recompile it for the QL using C68 'C' Compiler.
This program is restricted to Non-Commercial use (see XLISP_MEM 1.0 INTRODUCTION)
An Expanded QL with TK2 and at least one Disk Drive is required to run the program.
All Source Code is included on the Disk.

SPECIAL 4: QPACer (QPAC2 Boot Creator)

This program has been written to help those who have trouble in creating their own Boot programs to run QPAC2. QPACer asks plain and simple questions, it will then write a Boot program to set up QPAC2 just how you specified it. There is an extensive Manual in Quill dock, Perfection _per and ASCII –txt format.
Requirements are QL, 256K, QPAC2 & TK2.

Copying Fee:
£0.50p-User Supplies Disk £1.00p-Pre-Copied Disk £0.50p-User Supplies Disk £1.00p-Pre-Copied Disk.

2nd User H/ware & S/ware

2nd User Hardware and Software is available from time to time. Please ring for details of the current stock being held. Items available include as follows:-

QL's (Full & Backup), Disk I/Face's, Memory Expansion, Disk Drive's, Printer's, Monitor's (Colour & Green Screen), QL Related Books, QL User/World Back Issues, QUANTA Back Issues, Blank Disks's and all sorts of Commercial Software including Games and Utilities.

Public Domain Authors

QUBBESoft P/D are always seeking Authors of any type of Public Domain Software for the Q.L. If you have written a program for the Q.L. then why not send it in for evaluation.

Payment

Cheques/Postal Orders payable to :- 'K. Dunnett'

Postage & Packing

C68-50p (UK) £1.50 (EUROPE) £2.50 (REST OF WORLD)
EMACS, SPEC3, SPEC4-30p (UK) 70p (EUROPE)
£1 50 (REST WORLD)

38, Brunwin Road, Rayne, Braintree, Essex. CM7 5BU.
Tel No. (0376) 47852

CARE ELECTRONICS

QPAC II by Tony Tebby

MAKES YOUR QL TRULY MULTI - TASKING

QPAC II Main menu windows adjust automatically to size. Files, directory, view, print, delete, backup, jobs, pick, Rjob, sort, channels, things exec, wake, buttons, Hotkey, Hotjobs. Fully multitasking, allows many programs to run at once. Requires min of 256k expanded memory ..... @29.61b

SPECIAL OFFER ONLY £39.48b

Upgrade QRAM to QPAC II return master ..... @29.61b

PLEASE SEND SAE FOR A COPY OF THE QPAC II REVIEW

QPOWER REGULATOR

Switch mode power supply, QL runs cool, stops lock ups (due to voltage drop). Helps filter noise out. Simple to fit (no soldering) £25.85c

QL SUPERTOOL KIT II by Tony Tebby

Over 118 Commands: Full Screen Editor, Key Define Print Using, Last Line Recall, Altkey, Job Control, File Handling, Default Directories, Extended Network.
16k Eprom Cartridge Version ..... @£23.50d
Configurable Version of Microdrive ..... @£23.50d

ZITASOFT SOFTWARE by Steve Jones

LOCKSMITHE copies M/DRIVE - M/DRIVE ..... @£14.10c
4MATTER + LOCKSMITHE copies M/DRIVE - DISC ..... @£23.50c
SIDEWINDER - High resolution printer driver prints full screens or parts of screens from postage stamp size to large banners. Prints sideways, invert, scale, mirror, text insertion ..... @£20.21c
SIDEWINDER PLUS - (for expanded QL's) includes all the features of above, PLUS multiple label printing, desktop publishing files and printer drivers for The LC10 JX80 24 pin and colour printers (Please state 3.5" disc or M/Drive) ..... @£23.97c
Upgrade to Sidewinder Plus ..... @£8.46c

HOW TO ORDER

By Post. Enclose your Cheque/PO made payable to CARE Electronics or use ACCESS/VISA. Allow 7 days for delivery

SOFTWARE 87 (State MDV or Disc)

TEXT87 Plus 4 ..... @ £79.00b
FOUNTED 89 ..... @ £15.00
FOUNTED 88 ..... @ £25.00c
248B PRINTER DRIVER ..... @ £15.00c
Upgrade to Text 87 Plus 4 from V3.00. Return master £39.00b
Upgrade to Text 87 Plus 4 from all other versions. Return Master £49.00b
Send SAE for further info on above.

MIRACLE SYSTEMS PRODUCTS

0K Disc Interface (inc Tool Kit II) ..... @ £99.64b
QL Centronics Printer interface ..... @ £29.61d

QL HARDWARE

QLKeyboard Membrane ..... @ £11.75d
Care Eprom Cartridges each ..... @ £6.11c
ULA CHip ZX8301 ..... @ £15 98c

READY MADE LEADS

RGB QL TO PHONO £6.11c
RGB 8-6 PIN £7.52c
RGB 8-7 PIN HITACHI £7.52c
RGB 8-7 PIN FERGUSON £7.52c
RGB 8 PIN TO SCART £11.75c
RGB 8PIN TO CM8833 MK II £11.75c
6WAY PCC TO 25WAY D £9.87c

TONY TEBBY SOFTWARE (QJUMP)

QTYP Type/Spell Checker ..... @ £30.55d
QPAC 1-Desk top accessories, calendar, alarm, calculator, typewriter, digital clock sysmon ..... @ £22.56d

CARE ELECTRONICS
DEPT QL, 15 HOLLAND GDNS
GARSTON, WATFORD,
HERTS, WD2 6JN.
Tel: 0923-894064
Fax: 0923-672102

Please add carriage
a=£6.11 b=£3.76 c=no charge d=£2.35

Inside the

Simon Goodwin checks out the seven-disk C68 programming public domain C-system.

The C68 programming system is an important QL release. It is comprehensive, and it is free. It spans seven disks, including source code so that you can customise and re-compile almost the entire system. The 1992 version 2 is reputedly bigger still, and even better.

This article is based on several weeks' experience of C68 version 1.05, plus news of the update. C68 is complete, conforming to the loose K&R standard and approaching the latest ANSI C standard. Pre-written libraries support all Qdos traps, and the many standard C functions.

You can use the C68 to write fast, powerful QL-specific programs, or change libraries to get portable code using 'glass teletype' windows, that you can re-compile on any other micro with a C compiler, from PC, ST, Mac, and Amiga to Sun, Next and (D)RS work stations and beyond.

The documentation includes about 150 A4 pages in 20 *Quill* _DOC files, plus a compressed 200K language tutorial. The preset document page length may be 64 or 66 lines, and needs adjusting for fanfold A4 paper. In Quill or *Xchange* type F3 DP, then 70 or the length of your pages.

C68 users require a QL with QJump's *Toolkit 2*, at least 384K of ram and one or more 720K disk drives. Extra space is handy, but not essential unless you want to re-compile everything, unlike most MS– DOS C compilers, which are useless without a hard disk.

Before C

The precursor to C was BCPL (Basic Combined Programming Language) invented by Martin Richards at Cambridge University. BCPL was used to implement the Tripos operating system that spawned AmigaDos. BCPL is a simple language, with only one data-type, the word, making compilers easy to write and convert. It was the first compiled language implemented on the QL by Metacomco founder Tim King, and powered 1984 releases like *QL Assembler*, later renamed *Assembler for the QL* at the insistence of Sinclair's Intellectual Property department. QL BCPL generates slow double-offset 32 bit instructions and was soon eclipsed by more flexible compilers.

The C story began in 1970 when Ken Thompson, working at Bell Labs, USA wrote a compiler for a variant of BCPL, Known as B, as part of the development of the operating system Unix.

B begat C, designed by Dennis Ritchie and loosely defined in the book *The C Programming Language* by Kernighan and Ritchie, and refined in P J Plauger's *Standard C*. Like Tripos, but unlike most earlier systems, Unix is largely written in a high-level language. For more than a decade Unix and C have been held up as the systems of the future.

The C compiler allows large, complicated programs to be written piecemeal, while permitting low-level bit-twiddling operations that might otherwise need assembler code. C is much more portable than assembler, as one 'source' file can be compiled into 'object code' for any processor with a full C compiler. Unix versions proliferated in the 80's, but are coming together again now, after 20 years of development. Every Unix system consists of thousands of little C routines working together. Many fine, well-supported C programs are in the Public Domain, thanks to their origins in the educational promotion of Unix.

The first QL programs were written in C; Psion used a cross-compiler from their VAX 11-780 minicomputer. Later Quill et al were re-written in assembly code, shedding bugs, improving speed and reducing program size for Version 2.

In 1990 PDQL advertised a superior C system based on the later ST version of Lattice C with new libraries, but this was never released. Many of the features of *PDQC*, such as the linker and libraries, wound up in C68, along with other later developments.

I shall illustrate that compiled C programs are larger and slower than hand-crafted machine code to do the same job, but easier to maintain across many systems. C68 is superior to BCPL or earlier QL C compilers, and generates suprisingly efficient code, if you tell it the right things.

The C language translator is usually split into four tasks: pre-processor, parser, assembler and linker. Like Unix systems, C68 has a control program, CC, which calls the others as required.

Task and data

Version 1 of C68 stores tasks in two parts of the transient program area, like Quill. One holds the active task code, and the other is a data area. The amount of stack and heap memory available as a program runs is determined by a parameter option when it is loaded:

```
EX FLP1_CPIO;"%200 =50"
```

The **%** prefixes the heap requirement (200K) and = marks the stack limit, both in kilobytes. The parameter string may include file names or other options up to 32K in length, and channels and pipes may also be passed to a task. The default is 8K of heap space, 2K for the stack of local variables and parameters, and 4K for each pipe.

The new C68 Version 2 can allocate stack and heap space dynamically from the common heap. C68 tasks are not re-entrant, so you need one copy of the code for each task you run, but a shareable library is promised.

All but the tiniest C programs are usually written as several files, later linked into one task. This reduces the amount of re-compilation that is needed after changes to the source, and avoids the temptation to re-invent the wheel. The **Make** utility reads file dates to determine when files have changed. To work properly this needs the extended dates supported by Miracle's hard disk, Gold Card and the ST/QL emulator.

The pre-processor CPP reads source and strips out comments, replaces names with constant values and selects conditional code. The header file NICE_H includes CPP definitions to replace C brackets with meaningful words like WHILE, REPEAT and ENDIF, but sadly few programmers use it. **Listing one** shows the format of a typical C68 library function, STRTOL, which scans text for a number and returns a corresponding binary value.

The function has three parameters: NPTR, the address of the text in memory, ENDPTR, returned with the address of the last character scanned, and BASE, the number base from 2 (binary) to 36, using digits and letters A-Z. If BASE is initially 0, the actual base is determined from the text: a leading zero signifies octal, base 8, while 0x and 0X mark the start of hexadecimal values.

Lines starting with a hash are processed by CPP. It replaces the first line #INCLUDE <CTYPE.H> with the text of the file CTYPE.H, which further #INCLUDEs the three-line file TYPES_H, which calls up SYS_TYPES_H. These define constants and macros for later use.

STRTOL accumulates a 32 bit 'long' integer value, but the 68000 cannot directly multiply two 32 bit numbers, as I explained in *DIT Toolkit,* August 1991. _TEN_MUL and _BASEMUL are macro definitions

C68 Compiler

which replace constant multiplications with additions and << binary shift operations, avoiding calls to the slow library routine .LMUL. CPP replaces later instances of these names with the code they define.

There are serious but subtle bugs in listing one, but these show up rarely as most calls from other parts of the system let STRTOL infer the number base. These particular bugs may be fixed in Version 2, but they provide a good example of the general strengths and weaknesses of C.

CPP generates temporary files which are read by the parser. The parser C68 analyses the pre-processed text and translates it into assembly language. C68 comes with routines to generate 68000 and Intel 386 code. The libraries assume the 68000 or its QL cousin the 68008, and no 386 assembler is included.

The parser comes from Minix, the 'tiny' Unix system published in book form by Prentice-Hall. It generates good code if you set the right REGISTER variables and four pointers at once. With A6 and A7 pointing at stacks, this leaves only four

```
* C68-generated assembler output
* Annotations by Simon N Goodwin
.text
.even
strtol:
link    a6,#-10         Parameters
movem.l d3/d4/d5/d6/d7/a2/a3,-(a7)
move.l  8(a6),a2        Fetch NPTR
move.l  12(a6),a3       Fetch ENDPTR
move.l  16(a6),d3       Fetch BASE
moveq   #0,d5           RESULT = 0
moveq   #0,d6           NEGATIVE = 0
L1:                     WHILE
clr.w   d0
move.b  (a2),d0         C = *NPTR
move.w  d0,d4
beq     L2              -> if null
clr.l   d0              ISSPACE(C)
move.w  d4,d0
move.l  d0,a0           Make index
addq.l  #1,a0           1-256 from C
add.l   #_ctype,a0      Find table
move.b  (a0),d0
and.b   #8,d0           Test SPACE
beq     L2              -> not SPACE
addq.l  #1,a2           NPTR++
bra     L1              End of WHILE
L2:
clr.w   d0
move.b  (a2),d0         C = *NPTR
move.w  d0,d4
cmp.w   #43,d4          C == '+'
beq     L5              -> if true
cmp.w   #45,d4          C == '-'
bne     L3              -> if false
L5:
cmp.w   #45,d4          C == '-'
bne     L6              -> if false
moveq   #1,d0           1 if true
bra     L7              Set result
L6:
moveq   #0,d0           0 if false
L7:
move.w  d0,d6           Set NEGATIVE
addq.l  #1,a2           NPTR++
L3:
tst.l   d3              BASE == 0
bne     L8              -> if false
moveq   #10,d3          BASE = 10
cmp.b   #48,(a2)        *NPTR == '0'
bne     L10             -> if false
moveq   #8,d3           BASE = 8
addq.l  #1,a2           NPTR++
clr.w   d0
move.b  (a2),d0         C = *NPTR
move.w  d0,d4
cmp.w   #120,d4         C == 'x'
beq     L14             -> if false
cmp.w   #88,d4          C == 'X'
bne     L12             -> if false
L14:
moveq   #16,d3          BASE = 16
addq.l  #1,a2           NPTR++
bra     L11
L10:                    ELSE
cmp.l   #16,d3          BASE == 16
bne     L15             -> if false
cmp.b   #48,(a2)        *NPTR == '0'
bne     L15             -> if false
addq.l  #1,a2           NPTR++
cmp.b   #120,(a2)       *NPTR == 'x'
bne     L20             -> if false
moveq   #1,d0           1 if true
bra     L21
L20:
moveq   #0,d0           0 if false
L21:
move.w  d0,d4           C = 1 or 0 !
bne     L19
cmp.w   #88,d4          C == 'X'
bne     L17             -> Always!
L19:
addq.l  #1,a2           NPTR++
L17:
L15:                    }
L11:
L12:
subq.w  #1,a2           NPTR--
L22:                    WHILE
addq.l  #1,a2           NPTR++
clr.w   d0
move.b  (a2),d0         C = *NPTR
move.w  d0,d4
beq     L23             -> if null
clr.l   d0              ISDIGIT(C)
move.w  d4,d0
move.l  d0,a0           Make C into
addq.l  #1,a0           index, 1-256
add.l   #_ctype,a0      Find table
move.b  (a0),d0
and.b   #4,d0           Test DIGIT
beq     L24             -> not DIGIT
clr.l   d0
move.w  d4,d0           Fetch C
sub.l   #48,d0          Subtract '0'
move.w  d0,d7           Set DIGIT ;
bra     L25
L24:                    ELSE
clr.l   d0
move.w  d4,d0           Fetch C
move.l  d0,-(a7)        Stack it
clr.l   d1              ISUPPER(C)
move.w  d4,d1
move.l  d1,a0           Make C index
addq.l  #1,a0           1-256
add.l   #_ctype,a0      + Table base
move.b  (a0),d1         Extract byte
and.b   #1,d1           Extract BIT
beq     L26             -> Not UPPER
moveq   #65,d1          ? 'A'
bra     L27
L26:
moveq   #97,d1          : 'a'
L27:
move.l  (a7)+,d0        Unstack C
sub.l   d1,d0           - 'A' : 'a'
add.l   #10,d0          + 10
move.w  d0,d7           Set DIGIT ;
L25:
tst.w   d7              IF DIGIT < 0
blt     L30             -> negative
ext.l   d7              ;! DIGIT
cmp.l   d7,d3           >= BASE
ble     L23             if so, BREAK
L28:
cmp.l   #10,d3          BASE == 10
bne     L31             -> otherwise
move.l  d5,d0           Use _TEN_MUL
add.l   d0,d0           for * 10.L
add.l   d0,d0           D0 is D5 * 4
add.l   d5,d0           D0 is D5 * 5
add.l   d0,d0           D0 = D5 * 10
bra     L32
L31:
cmp.l   #16,d3          BASE == 16
bne     L33             -> otherwise
move.l  d5,d0           Prepare << 4
asl.l   #4,d0           D0 = D0 * 16
bra     L34
L33:
cmp.l   #8,d3           BASE == 8
bne     L35             -> otherwise
move.l  d5,d0           Prepare << 3
asl.l   #3,d0           D0 = D0 * 8
bra     L36
```

Inside C68 the Compiler

```
L35:
move.l  d5,-(a7)        Stack X
move.l  d3,-(a7)        Stack B
jsr     .lmul           The hard way
addq.l  #8,a7           Tidy stack
L36:                    Base 8 done
L34:                    Base 16 done
L32:                    Base 10 done
move.l  d0,d5           Set RESULT
ext.l   d7
add.l   d7,d5           Add DIGIT
bra     L22             End of WHILE
L23:
L30:
L8:
tst.w   d6              IF NEGATIVE
beq     L37             -> if false
move.l  d5,d0           Get RESULT
neg.l   d0              Negate value
move.l  d0,d5           Set RESULT
L37:
move.l  a3,d0           Test ENDPTR
beq     L39             -> if NULL
move.l  a2,(a3)         Set *ENDPTR
L39:
move.l  d5,d0           D0 is RESULT
L41:                    Restore regs
movem.l (a7)+,d3/d4/d5/d6/d7/a2/a3
unlk    a6              Lose params.
rts                     Return D0
.globl strtol
```

```
QL World March 1992, C library STRTOL source, page 1 of 1

#include <ctype.h>

/* defines to avoid long muls on a lowly 68k */
#define _TEN_MUL(X)     ((((X) << 2) + (X)) << 1)
#define _BASEMUL(B, X) \
    (((B) == 10) ? _TEN_MUL((X)) : (((B) == 16) ? ((X) << 4) :
    (((B) == 8) ? \ ((X) << 3) : ((B)*(X))  )))

/*********************************************************
 * Routine to use as base for all atoi, atol etc routines.*
 * Used by scanf etc.                                     *
 *********************************************************/

long strtol(nptr, endptr, base)
register unsigned char *nptr;
register char **endptr;
register int base;
{
  register unsigned short c;
  register long result = 0L;
  register short negative = 0;
  register short digit;

  while ((c = *nptr) && isspace(c))      /* skip leading white space */
        nptr++;

  if ((c = *nptr) == '+' || c == '-') { /* handle signs */
        negative = (c == '-');
        nptr++;
  }

  if (base == 0) {                       /* determine base if unknown */
        base = 10;
        if (*nptr == '0') {
                base = 8;
                nptr++;
                if ((c = *nptr) == 'x' || c == 'X') {
                        base = 16;
                        nptr++;
                }
        }
  else if (base == 16 && *nptr == '0') {
                nptr++;          /* discard 0x/0X prefix if hex */
                if ((c = *nptr == 'x') || c == 'X')
                        nptr++;
  }

  nptr--;                                /* convert the number */
  while (c = *++nptr) {
        if (isdigit(c))
                digit = c - '0';
        else
                digit = c - (isupper(c) ? 'A' : 'a') + 10;
        if (digit < 0 || digit >= base)
                break;
        result = _BASEMUL(base, result);
        result += digit;
        }
  }
  if (negative)
        result = -result;

  if (endptr != NULL)                    /* point at tail */
        *endptr = (char *)nptr;
  return result;
}
```

registers for the compiler's own templates.

Listing two is the parser output that corresponds to Listing one, with my annotations alongside, to show what's going on. D0, A0 and occasionally D1 are used for temporary results; if more space is needed, values are pushed onto the stack, as between L24 and L25. Prefix CLR instructions are widely used to clear the high bytes of registers that will hold unsigned values.

LINK instruction

The LINK instruction reserves storage space for the local variables, offset on A6 like the parameters. In practice the space is never used, as all the variables in STRTOL are stored in processor registers. The default code to fetch and store every value at an offset on A6 is less efficient, but allows more variables.

Labels are numbered from L1; the compiler allocates numbers as it goes along. Some are unused in this case; L41 is only needed if the function has multiple RETURNs, and L38 and L40 never appear at all.

The global label _CTYPE points at 256 bytes with bits set to categorise the character codes 0 to 255. ISSPACE, ISDIGIT and ISUPPER generate code to look up a code and test the relevant bit in the table. The address of _CTYPE is converted into an absolute address after the program is loaded. The ADDQ.L #1 instructions could be eliminated if _CTYPE pointed one byte later.

L8 turns up rather late, and reveals a bug in the source. The second WHILE loop that scans digits is conditional to the test IF (BASE==0). The code never scans any digits if an explicit base is specified. It's hard to see this from listing one, because of the erratic indentation of the source.

Another subtle bug crops up between L10 and L19. The closing bracket in IF ((C = *NPTR == 'x') should be before the == comparison. As written the code assigns the value of *NPTR == 'x' to C, giving a true or false result, 1 or 0. The code fails if the number starts 0X.

Register variables

Register variables save time, but C68 still has to transfer them into D0 to perform calculations. The three lines before L37 could be replaced with NEG.L D5, but the code generator does not spot this improvement.

The last three lines of source generate just four instructions, then C68 copies the result from D5 into D0, restores the old register values, un-links the local space and returns to the caller.

The lines that start with dots are assembler directives. The supplied assembler is unfriendly and limited, but it works. It is not

ideal for human programmers but suits the linker and assembly code subset generated by C68. The linker is reputedly three times faster than the GST linker supplied by Sinclair and Metacomco, but greedy for ram.

File utilities supplied with C68 include routines to maintain libraries and ARC, which can compress or restore groups of files in PC or QL format, with task headers. DIFF compares text files to find matching sections, but source for DIFF is not provided.

Phil Spink has written a public domain C68 task to read and write Unix CPIO disks on a 720K drive. With *Multi-DiscOver*, now marketed by Dilwyn Jones, the QL can expect a flood of C in future.

Programmers need a text editor to write C source. *Micro Emacs* is the obvious PD choice for an expanded QL; I plan to explore that in a future article. Other programming editors such as *ArcEd*, *Devpac* and *Spy The Editor*, are suitable.

In such a big system there are bound to be a few rough edges, and floating-point arithmetic was dodgy on versions up to 1.05. The current library uses 32 bit (7 digit) decimals, but 64 bit 15 digit resolution is promised for version 2.

There are some out-of-date object files in version 1.05. For instance the functions to convert QL and Unix universal dates give the wrong result by six hours, unless you re-compile LIBC_TIME_C.

Some bugs in the libraries stem from a lesser degree of testing than you might expect of a commercial product. Routines may work well in practice, but not as documented. All this can be fixed, as you have the source to re-compile.

Watch your step

The C language does not tolerate fools gladly. Code written for pure efficiency is often dangerous and non-portable. The compiler is simple-minded, and putting garbage in breeds garbage out. The package does not yet include LINT, a Unix tool that highlights machine-dependent or implausible code. Debugging facilities are available if you re-compile your program to include special signposts and error-trapping routines. Options allow single-stepping, watch-points and checks for memory overflow.

The separation of parts means that diagnostic messages pop up all over the place, and they may be hard to relate to the source. It is easy for faulty programs to corrupt memory outside the task. This is often fatal for the entire Qdos or MS-DOS system; on Unix it only knocks out the task.

Much of Qdos, with the notable exception of the file system, is modelled in style on Unix. The many parts of C68 may be daunting at first, but they provide a good grounding, and Unix enthusiasts will get quick results.

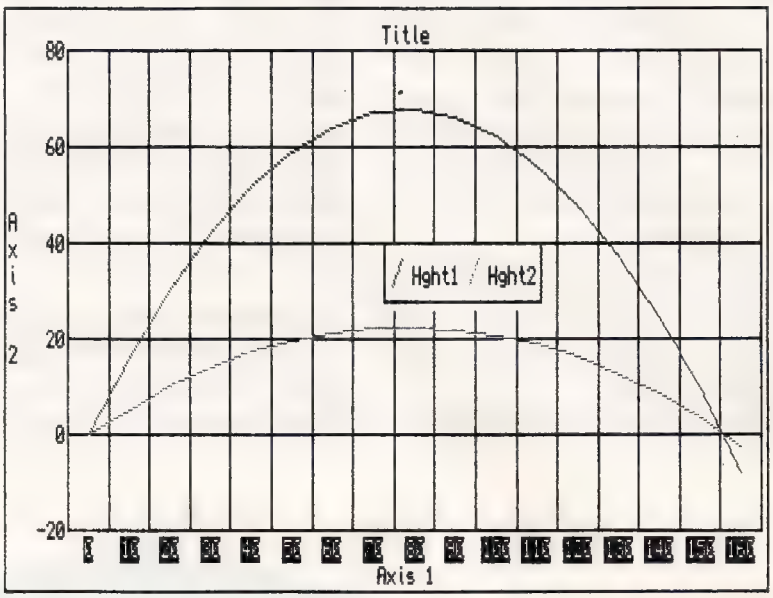
Real Unix systems rely on monsterous hard disks and virtual memory, but many of their programs will now run on the QL, thanks to the conversion efforts of Dave Walker and Jeremy Allison. C68 means that new and interesting QL programs can be developed with a prospect of later conversion and sale on other machines. This will extend the life of the QL and encourage programmers to continue to use it.

I have only skimmed the surface of C68 here, and hope to explore floating-point and function calling in a future article. As usual, comments from readers are welcome, care of *QL World*.

C68 is widely available from QL PD suppliers, including CGH Services (£2.00 per disk), Qubbesoft (£5.50 for all 7 disks in the UK, £6.50 for continental Europe or £7.50 anywhere in the world). Quanta members can have it for free, if they send blank disks to the C Librarian, Norman Dunbar. Try it!

New Uses For ABACUS

Howard Clase continues with the second part of his Abacus thesis.



If I haven't convinced you of the advantages of using Abacus rather than SuperBasic for a single calculation, consider the kind of problem that involves repeated application of a single formula with a varying parameter – for example the height of a missile at various distances during its flight. This is shown in **Listing one.**

AUTO_CALCULATE is turned off (<F3>, <D>, <A>, <ENTER>) so that the velocity and angle data can be altered easily in rows 9 and 10. The X values (horizontal distances) are generated by the formula in B14: row( ) returns the numerical value of the row and you have to subtract the number of first row (here 14) so that you start at the right place; the multiplier – in this example 25 – sets the increment between values. I set the increment large in my example to avoid filling the pages of *QL World* with too many meaningless numbers, and chose a smaller value and a longer table for better resolution. Columns C and D contain the formula for calculating the height (Y) as a function of distance (X) (in ordinary algebra this is:

y = x*tan(a)-g/2/v^2*x^2/(cos(a))^2,

where a, v, and g are angle, velocity and gravity respectively. This simple formula doesn't allow for wind resistance). Once you have typed this into C14 you can ECHO (<F3>,<E>) or COPY (<F3>, <C>) it to D14, but you then have to AMEND (<F3>, <A>) it to change the absolute references. Abacus is able to handle correcting relative cell references when formulae are copied around the grid – and even when extra rows or columns are inserted, but absolute values are left alone. In this instance all the $C references have to be changed to $D. Once you have got the first row of your table working you can generate columns as long as you like (memory permitting) in two ways: either by using the ECHO command (<F3>, <E>) or (my preference) by AMENDing (<F3>, <A>) the cell to read: col = [old formula], then press <ENTER> and put in the range you require. This should be done for columns B, C, and D. You can of course use any formula of your own devising and set up as many parallel columns as you need. The grid has also been prettied up a bit by reducing the column widths (<F3>, <G>, <W>) TO 7 (A) OR 6 (B-E), and the numbers set to integers (col B) or two decimal places (cols C and D) using the UNITS command (<F3>, <U>, etc.) Incidentally for an instant report of the current width of a column, type WIDTH( ) into a blank cell in the column (don't forget the brackets).

Graphs are best

Unless you have an amazing head for figures columns of numbers are not very informative; this sort of data is best presented in analogue form by way of a graph. You can plot a crude kind of graph in Abacus, but you get a much better picture if you export the data to Easel. The same approach can be used to produce a graph of any mathematical function.

I am not aware of a currently available commercial graph plotting program, but there have been at least three good amateur SuperBasic programs written: two

```
<<< Traj_aba >>>

   |   A   |   B   |   C   |   D   |  E
 1 |        Missile Trajectory
 2 |
 3 |^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 4 |<<<   Move the cursor to the
 5 |appropriate  cell  to  enter-
 6 |value.  Then press <F3>,<X>.
 7 |^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
 8 |
 9 |Velocity (m/s)      42      42 <<<
10 |Angle (deg)         60      30 <<<
11 |Gravity (m/s^2)    9.8     9.8
12 |
13 |            Dist   Hght1   Hght2
14 |               0    0.00    0.00
15 |              25   36.36   12.12
16 |              50   58.82   19.61
17 |              75   67.40   22.47
18 |             100   62.09   20.70
19 |             125   42.90   14.30
20 |             150    9.81    3.27


          FORMULAE

B14 (row()-14)*25

C14 B14*TAN(RAD($C10))-$C11/2/
    $C9^2*B14^2/COS(RAD($C10))^2

D14 B14*TAN(RAD($D10))-$D11/2/
    $D9^2*B14^2/COS(RAD($D10))^2
```

are available from the Quanta library, but the Microdrive Exchange version doesn't seem to have survived the coup! (As the author of one of them any further review from me is inappropriate.) If you have not got access to one of these then the Abacus/Easel method can give good results. The designer of the export/import interface between Abacus and Easel clearly had financial spreadsheets most in mind, so it is not as convenient for our present purpose as it might be, but as long as you keep to the rules it works quite well.

In the following discussion I shall assume the usual labelling of the axes, ie the Y axis is vertical and the X horizontal and also that the data are arranged in columns rather than rows.

Easel limitations

The first drawback is that Easel really only plots a series of Y values assuming that the X intervals are equally spaced. You can *label* the X axis with the values, but Easel cannot respond to the values; it just treats them as text. This means that you cannot increase the 'resolution' in areas of special interest; your table of X values must be evenly spaced. I find the easiest way to do this is to use the ROW( ) function to generate the X value as illustrated in Listing one.

The next problem is that your EXPORTed data block must have a TEXT label at the head of each column. If not the column is either ignored altogether or the first item of data is replaced by a default label (eg COLB$). The column of X values has to be in text format too. Fortunately it is quite easy to convert a column of values to text. To get a good graph set the increment so that there are at least 15 rows of values of interest in your table, and save your Abacus program to disk or mdv, since it is easier to re-LOAD it than undo the following. Then make sure that AUTO- CALCULATE is off and AMEND the formula in the first cell of the column (B14 here) as follows: add 'col=str(' before the existing formula and ',2,0)' after it (don't type the quotes!) so that it reads: col=str (old-formula,2,0). (The 2 selects integer format and the 0 is a dummy – see the 'str' function in the guide for more information.) Press <ENTER>, and tell Abacus which rows you want to extend your alteration to. This will replace all the numerical entries in the column with the equivalent character string. If you have done it right (and you haven't altered the justification defaults) all the numbers will move across to the left-handsides of their cells. Nothing else in the grid should change. Now EXPORT your block of data to Easel (<F3>, <F>, <E>, <E>, block reference) making sure that the block includes the text labels. In the example the block would be B13:D20. If you give the file a name like 'Traj' it will appear on your default device as 'Traj__exp'.

Now move to Easel. Select the appropriate display format – 3 is a conventional graph – by <F3>, <C>, <F>, <3> and IMPORT (<F3>, <F>, <I>, "Traj") to see your graph. In this instance you will get two trajectories for 60° and 30°, and demonstrate that the ranges are both the same, but that you would be better advised to use 60° if there are tall trees between you and your target. Of course you can dump this to your printer too; **Figure one** shows the output I got from 'Traj__aba' using an increment of 10.

```
100 nm$=                 "Aba_Eas_boot"

110 Q_tskf "mdv1_abacus",160: CLEAR
120 Q_tskf "mdv2_easel",64: CLEAR
130 REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
140 DEFine PROCedure            Q_tskf(task_name$,sp$)
150 REMark see: Simon Goodwin QL World April 1989 p 26
160 LOCal sys_vars,task_tag,ch_bas,last_ch_num,ch_ptr
170 sys_vars = 163840
180 space = "0"&sp$: DIM dummy$(space-1,1022)
190 IF Sys_W(96)=0: POKE_W sys_vars+96,1
200 task_tag = Sys_W(96): EXEC task_name$
210 PAUSE 200
220 ch_bas = Sys_L(120): last_ch_num = Sys_W(114)
230 FOR ch_ptr = ch_bas TO ch_bas+last_ch_num*4 STEP 4
240   IF PEEK(ch_ptr)<> 255: REMark is channel open?
250   ch_def = PEEK_L(ch_ptr)
260   IF PEEK_L(ch_def+4) = PEEK_L(PEEK_L(ch_bas)+4)
270     IF PEEK_W(ch_def+8) = task_tag
280       IF PEEK_L(ch_def)>120
290         POKE ch_def+67,1: EXIT ch_ptr
300   END IF :END IF :END IF :END IF
310   NEXT ch_ptr
320 PRINT\"Warning, no con channel in ";task_name$
330 END FOR ch_ptr
340 END DEFine
350 REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
360 DEFine FuNction                          Sys_L(ad)
370 RETurn PEEK_L(sys_vars+ad): END DEFine
380 REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
390 DEFine FuNction                          Sys_W(ad)
400 RETurn PEEK_W(sys_vars+ad): END DEFine
```

Switching

If you have Taskmaster or one of the other multitasking facilities switching between Abacus and Easel is easy. If you do not have a multitasking facility like Taskmaster or Qram and have a minimum of 256K extra memory you can use Simon Goodwin's *Taskforce (QL World* April 19 1989) modified as in **listing two.** This will work as a boot on its own, but you may, of course, add extra lines of your own eg to activate TK2 or Speedscreen or to set ALTKey definitions. I know it works with a JSU rom, and I think that it should work with other Sinclair roms, but cannot guarantee it. It will need modification to work with Minerva in two screen mode.

Lines 110 and 120 should be altered according to your system – eg if you use disks then it is easy to have both programs on the same disk. I happen to be writing this on my backup machine whose only adornment is an extra 256K. It requires a certain amount of juggling of mdv cartridges since both Psion programs are configured for mdv1, so the right one has to be there when required eg for help or printing. (You do not need to reconfigure them just to EXEC them from a different drive.) The second parameter is the amount of memory (K) *protected* (in the array dummy$ at line 180) so the *larger* the number the *smaller* the space available for your Psion program. The numbers given divide the spare memory about equally between the two Psion programs and SuperBasic for this amount of memory.

In operation you switch between the three by pressing <CTRL+C>; <SHIFT+F5> will restore your Psion windows, and you have to use the CLS#n command to clear the SuperBasic windows. If you have TK2 this is an obvious application for the ALTKEY function. You can of course use the routine to multitask any two EXECed programs – or more if you have enough memory. For more information on Taskforce see the original article (line 150).

If you do not have extra memory then you will have to reset and reboot with Easel.

Marks not money

Apart from the fact that the numbers represent marks rather than money this is very similar to a financial spreadsheet, but I include it because it illustrates the use of the rarely used 'lookup' function and also

contains a printing trick. The printout and formulae are shown in **listing three.**

Where a formula is used over a range of cells I give the range, and show the formula as it appears in the first cell. Since all the addresses are relative it can be easily copied to the other cells by using the COPY, ECHO, row=, or col= facilities as appropriate. The average formula in row 12 only occurs in alternate cells, the simplest way to achieve this is to echo to the whole block (D12:J12) and rubout (<F3>, <R>, <ENTER>) the cells you don't want.

The various marks are entered directly into columns D, F and H, and the total calculated in column J. We have to round off marks to the nearest 5 and letter grades are awarded as follows: A100-80, B75-65, C 60-55, D 50, F 45-0. The first is accomplished in cells L6:L10 using the usual rounding off formula and the letter grades determined from the lookup table in the block N12:O17. This requires a more detailed look.

Lookup table

The lookup table is just two columns of numbers, the first is the lowest mark for each grade in descending order, and the second the Ascii code of the corresponding letter. The lookup command is in three parts: the range to search, the offset of the column where it will find the answer, and the value to search for. Consider M6, the value to search for is found in L6 ie 80. The search starts at N6, and continues down until it finds the last cell containing a value no greater (equal is OK) than the search value of 80. It then returns the corresponding value found in the column offset by the specified amount. In this case the offset is 1 and the return value is 65, the Ascii value for the letter A. It is obviously important to have the values in your lookup list in descending order, and you must include both the minimum and maximum values in the table.

Lookup limitations

Unfortunately, the lookup function in Abacus is not very well thought out. The main problem is that only relative and not absolute addressing is allowed. This means that the command starts looking for the table one row lower each time it descends a row of the table. The only way around this is to put the top of the lookup table lower than the lowest cell that calls it and leave the column above free of any numerical values (luckily it ignores any text entires it may come across, so you could put some text in column N – in fact I do, but it doesn't appear on screen – see below). Lookup tables can be in rows as well as columns in which case turn all the above comments sideways.

The other point of special interest in this Abacus program is the invisible text in the upper part of column N. I find it is easy to lose my place when looking across wide tables of numbers, unless they are divided horizontally at regular intervals – every five is about right. The formula in N6:N10 is the way to do this on the printer; I have set the interval to three here to keep my example table small. The condition tests whether or not the row-number-plus-one is an exact multiple of three, if not it prints an empty string, and if it is it prints an Ascii 10, or line feed. This doesn't appear on the Abacus screen either, but it does have the desired effect on the printer as you can see. I usually use the ae diphthong character <CTRL+SHIFT+8> – Ascii 138 – instead of CHR(10), since my Epson printer treats this as a line feed too, and I can see where the blank line is going to come in advance. You'll have to test this out on your own printer.

Column widths

The bit at the bottom below the double line isn't really part of the program, but is just put in to show how I have adjusted the column widths. The two formulae in A20 and A21 both give text, and the block (A20:O21) is centrally justified.

In my introduction I mentioned that I had, on occasions, used Abacus to generate pure text output. This has been for the production of voting slips, tickets and other small items that require the use of a guillotine on regular sized pieces of paper. The trouble with *Quill* is that while you can copy whole lines vertically, you cannot copy part lines horizontally – but you can do this in Abacus.

Border pattern

The example shown in **listing four** is more or less self evident. It started with a 20 character x 11 line block A1:C11 – to fit neatly onto a 80 x 66 page (24 in all). Note that the formulae which generate the border pattern all refer absolutely to A2 (including A2 itself initially), get your text looking right in the prototype, and then copy it to fill a horizontal row with clones. Edit the serial number cell in the clones to increment the copy on its left by one, and then copy the rows down the page, changing the serial number of the first ticket on each new row to calculate its value from the one above so that they ultimately all refer back to the value in the prottoype.

Serial numbers

Once the page is complete you can quickly change the border character by inserting a new character into cell A2 and pressing <F3>, <X>, and you can change all the serial numbers by changing the value in B10. Of course anyone with a QL will be able to forge their own tickets, but they should be pretty easy to track down

```
                    <<<< Class_aba >>>>

  | A |       B        |C| D  |E| F  |G| H  |I|  J  |K| L  |M| N | O
 1|    Chemistry 2210                          Fall 1999
 2|
 3|          Name          Term    Labs    Final   Total    Mark/Gr
 4|                          20      20      100     100
 5|-------------------------------------------------------------------
 6|   1 Aardvark, Ethel  | 16.7 | 18.1 | 75.8 |  80.3 |    80 A
 7|   2 Bartok, Bella    | 10.5 |  2.2 | 36.0 |  34.3 |    35 F
 8|   3 Caesar, Julius   | 12.5 | 14.0 | 43.0 |  52.3 |    50 D

 9|   4 Dean, James      | 15.6 | 16.2 | 64.7 |  70.6 |    70 B
10|   5 Eliot, Thomas    | 10.5 | 11.5 | 52.0 |  53.2 |    55 C
11|-------------------------------------------------------------------
12|     Averages as %      65.8    62.0    54.3    58.1              0  70
13|                                                                 50  68
14|                                                                 55  67
15|                                                                 65  66
16|                                                                 80  65
17|                                                                100  33
18|===================================================================
19|                          Column Widths
20| A         B         C  D   E  F   G  H   I   J   K  L   M  N   O
21| 3        15         1  4   1  4   1  4   1   5   1  4   1  3   3


                          Key Formulae

    A5:M5,A11:M11   rept("-",width()+1)
    A6:A10          row()-5
    J6:J10          D6+F6+H6*0.6
    L6:L10          5*(int((J6+2.5)/5)
    M6:M10          chr(lookup(N6:N17,1,L6))
    N6:N10          if((row()+1)/3-int((row()+1)/3)>0,"",chr(10))
    D12,E12,J12,L12 ave(D6:D10)/D4*100

    A20:O20         chr(col()+64)
    A21:O21         str(width(),2,0)
```

– you probably know all the likely forgers in your area already!

The generation of attractive looking tables is also easier using Abacus than with Quill. Although Quill has a fairly flexible TAB command I find Abacus more useful here too for two reasons: it is much simpler to generate nice borders using "col = ", rept( ) etc (see my ticket example) and the option to ORDER the columns in alphabetical or numerical order is often useful.

Best calculator

In these two articles I have tried to demonstrate why many professional scientists have taken to spreadsheets as the best new calculation aid since the invention of the hand-held calculator. If you are a student or teacher in maths, physics or chemistry I hope I have given you some ideas as to how useful Abacus can be. In fact whatever you do, if you have just been using Abacus to balance your cheque book or even not at all I hope I have been able to convince you that it is worth a second look – don't let the accountants have all the fun! How about an Abacus game, someone?

<<<< Ticket_aba >>>>

	A	B	C	D	E	F
1	%%%%%%%%%%%%%%%%%%%%			%%%%%%%%%%%%%%%%%%%%		
2	%	New Millenium	%	%	New Millenium	%
3	%	Party	%	%	Party	%
4	%		%	%		%
5	%	December 31 2000	%	%	December 31 2000	%
6	%	10.00 pm	%	%	10.00 pm	%
7	%		%	%		%
8	%	Admit One	%	%	Admit One	%
9	%	$25	%	%	$25	%
10	%	1	%	%	2	%
11	%%%%%%%%%%%%%%%%%%%%			%%%%%%%%%%%%%%%%%%%%		
12	%%%%%%%%%%%%%%%%%%%%			%%%%%%%%%%%%%%%%%%%%		
13	%	New Millenium	%	%	New Millenium	%
14	%	Party	%	%	Party	%
15	%		%	%		%
16	%	December 31 2000	%	%	December 31 2000	%
17	%	10.00 pm	%	%	10.00 pm	%
18	%		%	%		%
19	%	Admit One	%	%	Admit One	%
20	%	$25	%	%	$25	%
21	%	3	%	%	4	%
22	%%%%%%%%%%%%%%%%%%%%			%%%%%%%%%%%%%%%%%%%%		

Formulae

```
A1:B1, A11:B11  rept($A2,width()+1)
A2:A10, C1:C11  $A2

E10             B10+1
B21             B10+2
E21             B21+1
```

Internal Number Chapter

Simon Wallis goes back into his numerical routines and adds division.

In my article of October 1990 I presented a suite of routines which carried out arithmetic operations on arbitrarily large integers. This article expands the list of available operations to include division, uses fixed point arithmetic instead of being restricted to integers, and explains how to calculate various trigonometric functions. Two programs give examples of correct use of the routines – the first evaluates the mathematical constant 'e', the base of Naperian logarithms; the second program is a four-function calculator.

Pencil and paper

The algorithms used for multiplication, addition and subtraction are essentially the same as in my October 1990 program, although all have been optimised and as such the assembly language listing is considerably longer. However the speed increase is well worth a few extra DATA statements. The algorithm used here for long division is based on the pencil-and-paper method we all learned at school. Although the algorithm is asymptotically not the quickest possible, it performs very well on reasonably small numbers, say up to a couple of hundred digits long. Because it is so long, we are not printing the assembly listing (**listing one**) but this will be sent to any reader who sends a large SAE and postage to *QL World*.

I chose to store a fractional number in the same way as I previously used to store an integer, but with a variable called

LISTING 2 - Machine code loader, and DATA statements

```
1800 REMark  Routine to load machine code from data statements into memory
1810 DEFine PROCedure SetUpMcode
1820 LOCal loop, Checksum, a,b,c,d,e,f
1830 PRINT 'Loading machine code from DATA statements'
1840 maxlength = 80 : decimals = 20
1850 accumulator = 0
1860 base = RESPR (1280 + 20 * maxlength): REMark space for 20 bignums
1870 addr = base
1880 RESTORE
1890 REPEAT loop
1900   IF EOF THEN EXIT loop
1910   READ a, b, c, d, e, f, Checksum
1920   IF a+b+c+d+e+f <> Checksum: PRINT 'Error in data': STOP
1930   POKE_W addr, a  : POKE_W addr+2, b: POKE_W addr+4,  c
1940   POKE_W addr+6, d: POKE_W addr+8, e: POKE_W addr+10, f
1950   addr = addr + 12
1960   PRINT '.';: REMark Some activity whilst loading machine code.
1970 END REPeat loop
1980 PRINT
1990 :
2000 DATA 18663, 3072, 18938, 1018, 31232, 14868, 87791
2010 DATA 51397, 18938, 1012, 55748, 19679, 48, 146822
2020 DATA 20085, 18663, 2056, 10241, 24832, 65502, 141379
2030 DATA 8780, 10242, 24832, 65494, 9292, 19679, 138319
2040 DATA 4112, 20085, 18663, 11400, 18938, 966, 74164
2050 DATA 31232, 14868, 10242, 24832, 65468, 8268, 154910
2060 DATA 10241, 24832, 65460, 6360, 20941, 65532, 193366
2070 DATA 19679, 4404, 28672, 20085, 18663, 18696, 110199
2080 DATA 18938, 926, 32256, 15892, 21383, 10241, 99636
2090 DATA 24832, 65426, 28672, 6336, 20943, 65532, 211741
2100 DATA 19679, 4242, 20085, 18663, 2056, 18938, 83663
2110 DATA 892, 28672, 12308, 21376, 10241, 24832, 98321
2120 DATA 65392, 18972, 26112, 6, 20936, 65528, 196946
2130 DATA 21120, 18938, 862, 14464, 19679, 4112, 79175
2140 DATA 20085, 49916, 40, 53934, 48, 45742, 169765
2150 DATA 52, 25198, 8310, 6144, 18938, 836, 59478
2160 DATA 32256, 15892, 31745, 18938, 824, 31232, 130887
2170 DATA 14868, 21381, 39559, 30463, 10242, 24832, 141345
2180 DATA 65320, 29184, 4636, 3073, 0, 26368, 128581
2190 DATA 4, 31744, 3078, 1, 26368, 10, 61205
2200 DATA 1537, 48, 28677, 20035, 20941, 65504, 136742
2210 DATA 3078, 0, 26368, 8, 29232, 28677, 87363
2220 DATA 20035, 29230, 28677, 20035, 10759, 21381, 130117
2230 DATA 29184, 4636, 1537, 48, 28677, 20035, 84117
2240 DATA 20941, 65522, 20085, 28922, 20085, 18663, 174218
2250 DATA 1912, 24832, 65258, 18426, 720, 31744, 142892
2260 DATA 18938, 710, 31232, 14868, 54213, 54725, 174686
2270 DATA 55237, 32256, 21381, 7713, 56866, 56838, 230291
2280 DATA 31744, 3079, 10, 27398, 31745, 1031, 95007
2290 DATA 10, 5895, 20941, 65512, 28672, 19679, 140709
2300 DATA 7904, 20085, 19074, 26368, 46, 18663, 92140
2310 DATA 27672, 18938, 648, 31232, 14868, 10241, 103599
2320 DATA 24832, 65150, 9804, 55234, 39554, 21381, 215955
2330 DATA 21378, 6363, 20941, 65532, 16924, 20938, 152076
2340 DATA 65532, 19679, 6198, 28672, 20085, 19074, 159240
2350 DATA 26368, 65528, 18663, 27672, 18938, 594, 157763
2360 DATA 31232, 14868, 10241, 24832, 65096, 10245, 156514
2370 DATA 39554, 55748, 9804, 38850, 21381, 21378, 186715
2380 DATA 6435, 20941, 65532, 28672, 6400, 20938, 148918
2390 DATA 65532, 19679, 6198, 20085, 18663, 27912, 158069
2400 DATA 10241, 10754, 29195, 24832, 65140, 30209, 170371
2410 DATA 29195, 24832, 65132, 8707, 29707, 24832, 182405
2420 DATA 65084, 8708, 24832, 65324, 29195, 29696, 222839
```

mbers two

```
2390 DATA 65532, 19679, 6198, 20085, 18663, 27912, 158069
2400 DATA 10241, 10754, 29195, 24832, 65140, 30209, 170371
2410 DATA 29195, 24832, 65132, 8707, 29707, 24832, 182405
2420 DATA 65084, 8708, 24832, 65324, 29195, 29696, 222839
2430 DATA 24832, 65070, 21123, 3075, 11, 26112, 140223
2440 DATA 65506, 29195, 24832, 65094, 32256, 18938, 235821
2450 DATA 484, 15892, 21383, 10245, 24832, 64986, 137822
2460 DATA 5148, 21122, 29195, 24832, 65274, 29696, 175267
2470 DATA 24832, 65022, 29697, 24832, 65328, 20943, 230654
2480 DATA 65512, 18938, 446, 29696, 13332, 29184, 157108
2490 DATA 24832, 65364, 28672, 19679, 4278, 20085, 162910
2500 DATA 18663, 3336, 18938, 418, 31232, 14868, 87455
2510 DATA 21381, 10241, 24832, 64918, 55749, 7700, 184821
2520 DATA 3079, 9, 26112, 16, 28672, 6272, 64160
2530 DATA 21388, 20941, 65518, 24576, 6, 20999, 153428
2540 DATA 6279, 19679, 4272, 28672, 20085, 18663, 97650
2550 DATA 7176, 18938, 360, 31232, 14868, 21381, 93955
2560 DATA 10241, 24832, 64860, 6164, 30217, 38404, 174718
2570 DATA 6339, 20941, 65526, 24832, 65444, 19679, 202761
2580 DATA 4152, 28672, 20085, 18663, 24832, 11777, 108181
2590 DATA 29184, 24832, 64876, 24832, 65478, 9223, 218425
2600 DATA 24832, 65112, 19679, 134, 28672, 20085, 158514
2610 DATA 18663, 1128, 24832, 64824, 18938, 282, 128667
2620 DATA 31232, 14868, 21381, 45834, 28160, 10, 141485
2630 DATA 27904, 12, 20941, 65524, 28673, 24576, 167630
2640 DATA 4, 28672, 18938, 248, 14464, 19679, 82005
2650 DATA 5664, 20085, 18663, 1128, 24832, 64772, 135144
2660 DATA 18938, 230, 31232, 14868, 21381, 45834, 132483
2670 DATA 28160, 14, 27904, 16, 20941, 65524, 142559
2680 DATA 24576, 8, 28673, 24576, 4, 28672, 106509
2690 DATA 18938, 192, 14464, 19679, 5664, 20085, 79022
2700 DATA 18663, 24840, 11778, 9217, 29185, 24832, 118515
2710 DATA 64736, 18938, 170, 29696, 13332, 24832, 151704
2720 DATA 65036, 9223, 29186, 24832, 64716, 29185, 222178
2730 DATA 24832, 64784, 11776, 29186, 24832, 64776, 220186
2740 DATA 40576, 29187, 24832, 64734, 29186, 29697, 218212
2750 DATA 24832, 65418, 19072, 26112, 98, 29188, 164720
2760 DATA 24832, 64714, 29188, 29698, 24832, 65346, 238610
2770 DATA 19072, 26112, 78, 29186, 9223, 24832, 108503
2780 DATA 64964, 29187, 29697, 24832, 64956, 29186, 242822
2790 DATA 29697, 24832, 65368, 19072, 26112, 22, 165103
2800 DATA 29187, 24832, 65172, 29185, 29698, 24832, 202906
2810 DATA 65266, 29696, 24832, 64610, 29186, 29697, 243287
2820 DATA 24832, 65334, 19072, 26368, 65490, 29186, 230282
2830 DATA 29697, 24832, 64954, 20943, 65470, 29184, 235080
2840 DATA 29699, 24832, 64576, 19679, 4230, 28672, 171688
2850 DATA 20085, 0, 50, 10, 0, 0, 20145
2860 SUBTRACT  = base +     702
2870 WRITENUM  = base +     170
2880 MULTIPLY  = base +     476
2890 DECIMAL   = base +    1026
2900 SHIFTLEFT = base +     364
2910 NEGATE    = base +     658
2920 INCR      = base +     600
2930 ADD_      = base +     298
2940 NUMBASE   = base +    1028
2950 COPYNUM   = base +      52
2960 MAXLEN    = base +    1024
2970 DIVISION  = base +     840
2980 SETTO0    = base +      92
2990 SHIFTRI   = base +     418
3000 POKE_W DECIMAL, decimals: POKE_W MAXLEN, maxlength
3010 END DEFine
3020 :
```

```
LISTING 3 - Evaluation of 'e' and sine(1)
Use in conjunction with listing 2


60 SetUpMcode
70 t=19: CALL SETTO0, t: CALL INCR, t: CALL SHIFTLEFT, t, decimals
80 BigSin t, 25
90 PRINT \'SIN(1) IS EXACTLY ';: CALL Writenum, 1, 0: PRINT
100 :
110 t=19: CALL SETTO0, t: CALL INCR, t: CALL SHIFTLEFT, t, decimals
```

DECIMALS determining where the decimal point should be placed; for example, the number 12.34 might be stored as 1234000 with a note that the last five digits should be to the right of the decimal point. Note that each number has the same number of decimal places, ie the variable DECIMALS is global, and if it is changed, then the setting-up routine (**listing two**) must be run again.

The addition and subtraction routines needed no modification to operate correctly on fixed-point numbers, as 1234000 + 1234000 = 2468000, which we can read as 12.34 + 12.34 = 24.68. However, more care was needed for multiplication, where the result must be shifted to the right, and division, where the result must be shifted to the left. Consider 12.34/12.34, which we know is 1. 1234000/1234000 is also 000001, which, given our machine representation, we should read as 0.00001. So the number of places to shift the result is the number of places that the implicit decimal point is from the right-hand end of the number. This is implemented so that the numerator is shifted left before the operation is carried out, and no check is made for overflow. Beware of this, and always make sure that MAXLENGTH and DECIMALS – the maximum number of digits of precision and the size of the fractional part respectively – are higher than you will need.

Big numbers

The machine code loader program (**listing two**) allows room for 20 'big numbers', numbered 0 to 19. Of these, 0 through 13 are used internally by the arithmetic routines, leaving you free to use 14 through 19 for your programs. The operations Add, Subtract, Multiply and Division each leave their result in big number # 0 – a sort of accumulator variable. The other routines – Copynum, SetTo0, Writenum, Shiftleft, Shiftright, Incr and Negate all operate 'in place', that is, they don't have any effect on the accumulator (unless of course the accumulator is supplied as one of the arguments).

If you ask your QL what the value of the mathematical constant 'e' is, for example:

type PRINT EXP(1)

you will get the result 2.718282.

How does it know? And can we improve the accuracy of the result? One way of programming the EXP function would be to pre-compute it for all values, and store the results in a look up table. A nice idea, but there are far too many possible values which it could be asked. A book of four figure tables might have sixty pages. The QL only has a 48K rom – not nearly enough to store details of sines, cosines, square roots, exponentials, logarithms, etc. One method of calculating EXP(X) is to

Internal Numbers chapter two

```
120 BigExp t, 25
130 PRINT \'e IS EXACTLY ';: CALL Writenum, 1, 0
140 STOP
150 :
160 DEFine PROCedure BigExp (X, iterations)
170 LOCal loop, one, N, Nfact, XpowN, Result, temp
180 one = 12
190 N = 13
200 Nfact = 14
210 XpowN = 15
220 Result = 16
230 temp = 17
240 CALL SETTO0, one
250 CALL INCR, one
260 CALL SHIFTLEFT, one, Decimals: REMark number one now equals 1
270 CALL COPYNUM, N, one
280 CALL COPYNUM, Nfact, one
290 CALL COPYNUM, XpowN, X
300 CALL COPYNUM, result, one
310 FOR loop = 1 TO iterations
320   CALL MULTIPLY, Nfact, N
330   CALL COPYNUM, Nfact, accumulator
340     REMark   Nfact = Nfact * N
350   CALL MULTIPLY, XpowN, X
360   CALL COPYNUM, XpowN, accumulator
370     REMark   XpowN = XpowN * X
380   CALL ADD_, n, one
390   CALL COPYNUM, n, accumulator
400     REMark   n = n + 1
410   CALL Division, XpowN, Nfact
420     REMark    accumulator = XpowN / Nfact
430   CALL COPYNUM, temp, accumulator
440   CALL ADD_, Result, temp
450     REMark    result = result + XpowN / Nfact
460   CALL COPYNUM, Result, accumulator
470 PRINT \ 'Loop=' ; loop ! 'Result=';: CALL Writenum, 1, result
480 END FOR loop
490 CALL COPYNUM, accumulator, result
500 END DEFine
510 :
520 DEFine FuNction EVEN (g)
530 RETurn (g + 1) MOD 2: END DEFine
540 DEFine FuNction ODD (g)
550 RETurn NOT (EVEN (g)): END DEFine
560 :
570 DEFine PROCedure BigSin (X, iterations)
580 LOCal loop, one, N, Nfact, XpowN, Result, temp
590 one = 12
600 N = 13
610 Nfact = 14
620 XpowN = 15
630 Result = 16
640 temp = 17
650 CALL SETTO0, one
660 CALL INCR, one
670 CALL SHIFTLEFT, one, Decimals: REMark number one now equals 1
680 CALL COPYNUM, N, one
690 CALL COPYNUM, Nfact, one
700 CALL COPYNUM, XpowN, X
710 CALL SETTO0, result
720 FOR loop = 1 TO iterations
730   CALL MULTIPLY, Nfact, N
740   CALL COPYNUM, Nfact, accumulator
750     REMark   Nfact = Nfact * N
760   CALL MULTIPLY, XpowN, X
770   CALL COPYNUM, XpowN, accumulator
780     REMark   XpowN = XpowN * X
790   CALL ADD_, n, one
800   CALL COPYNUM, n, accumulator
810   IF ODD (loop) THEN
820     CALL Division, XpowN, Nfact
830     CALL COPYNUM, temp, accumulator
840       REMark temp = XpowN / Nfact
850     IF ODD (loop DIV 2) THEN CALL Negate, temp
860     CALL ADD_, Result, temp
870     CALL COPYNUM, Result, accumulator
880       REMark Result = Result + XpowN / Nfact
890     PRINT \'Loop='; loop ! 'Result=';: CALL Writenum, 1, result
900   END IF
910 END FOR loop
920 CALL COPYNUM, accumulator, result
930 END DEFine
940 :
```

```
Sample output of listing 3,
showing convergence of power series


Loop=1 Result=1.0000000000000000000
Loop=3 Result=0.83333333333333333334
Loop=5 Result=0.84166666666666666667
Loop=7 Result=0.84146825396825396826
Loop=9 Result=0.84147100970017636684
Loop=11 Result=0.84147098464806798140
Loop=13 Result=0.84147098480865841976
Loop=15 Result=0.84147098480789370339
Loop=17 Result=0.84147098480789651484
Loop=19 Result=0.84147098480789650662
Loop=21 Result=0.84147098480789650663
Loop=23 Result=0.84147098480789650663
Loop=25 Result=0.84147098480789650663
SIN(1) IS EXACTLY 0.84147098480789650663

Loop=1 Result=2.0000000000000000000
Loop=2 Result=2.5000000000000000000
Loop=3 Result=2.66666666666666666666
Loop=4 Result=2.70833333333333333332
Loop=5 Result=2.71666666666666666665
Loop=6 Result=2.71805555555555555553
Loop=7 Result=2.71825396825396825394
Loop=8 Result=2.71827876984126984124
Loop=9 Result=2.71828152557319223982
Loop=10 Result=2.71828180114638447967
Loop=11 Result=2.71828182619849286511
Loop=12 Result=2.71828182828616856389
Loop=13 Result=2.71828182844675900225
Loop=14 Result=2.71828182845822974784
Loop=15 Result=2.71828182845899446421
Loop=16 Result=2.71828182845904225898
Loop=17 Result=2.71828182845904507043
Loop=18 Result=2.71828182845904522662
Loop=19 Result=2.71828182845904523484
Loop=20 Result=2.71828182845904523525
Loop=21 Result=2.71828182845904523526
Loop=22 Result=2.71828182845904523526
Loop=23 Result=2.71828182845904523526
Loop=24 Result=2.71828182845904523526
Loop=25 Result=2.71828182845904523526
e IS EXACTLY 2.71828182845904523526
```

use a power series – a formula which, if you add up enough of its terms, eventually converges to an approximate answer. It is impossible to obtain an exact value for e, or pi, as both belong to the class of transcendental numbers.

Values of e

One power series is EXP(X) = 1 + X + X^2/2! + X*3/3! + . . . + X^N/N!, where X^N means X raised to the Nth power, and N! means N factorial, the product of N with all the numbers less than it, ie N * (N-1) * (N-2) * . . . * 2 * 1. **Listing three** uses this power series to calculate an approximate value of e. By approximate I mean correct to perhaps eighteen places of decimals. Although twenty places are printed out, the last few may be wrong due to small rounding errors being accumulated gradually. If you want to work out more places than this, adjust MAXLENGTH and DECIMALS accordingly, and increase the number of iterations. Beware that for large numbers of iterations, N factorial gets very large, so ensure that MAXLENGTH is large enough to avoid (undetected) overflow. Also, make sure that MAXLENGTH is always more than twice DECIMALS, for the reasons explained earlier.

Similar power series exist for trigonometric functions, viz:

SIN(X) = 0 + X + 0 - X^3/3! +

```
LISTING 4 - CALCULATOR.
Use in conjunction with listing 2.


100 SetUpMcode
110 Intro
120 REPeat forever
130   key = CODE (INKEY$ (-1))
140   SELect ON key
150     = 48 TO 57                    : NumberPressed
160     = decpoint                    : PointPressed
170     = equals, enter               : EqualsPressed
180     = plus, minus, times, divide : OperatorPressed
190   END SELect
200 END REPeat forever
210 :
220 DEFine PROCedure Intro
230 MODE 8
240 PRINT 'FOUR FUNCTION CALCULATOR'
250 PRINT 'Maximum integer part  = '; maxlength - decimals; ' digits'
260 PRINT 'Maximum fraction part = '; decimals; ' digits'
270 accumulator = 0    : CALL SetTo0, accumulator
280 DisplayedNum = 12 : CALL SetTo0, DisplayedNum
290 CurrentNum = 13    : CALL SetTo0, CurrentNum
300 times  = CODE ('*')
310 divide = CODE ('/')
320 decpoint = CODE ('.')
330 minus  = CODE ('-')
340 equals = CODE ('=')
350 plus   = CODE ('+')
360 enter  = 10
370 LastOp = plus
380 INTEGER = -1
390 ShiftPending = INTEGER
400 END DEFine
410 :
420 DEFine PROCedure NumberPressed
430 IF LastOp = equals THEN
440   LastOp = plus
450   CALL SetTo0, DisplayedNum
460   CALL SetTo0, CurrentNum
470 END IF
480 IF ShiftPending = 0 THEN RETurn
490 PRINT CHR$ (key);
500 CALL ShiftLeft, CurrentNum, 1
510 POKE Numbase + Maxlength * (CurrentNum + 1) - 1, key - 48
520 REMark least significant digit of CurrentNum
530 IF (ShiftPending <> INTEGER) THEN ShiftPending = ShiftPending - 1
540 END DEFine
550 :
560 DEFine PROCedure PointPressed
570   IF ShiftPending <> INTEGER THEN RETurn
580   PRINT '.';
590   ShiftPending = decimals
600 END DEFine
610 :
620 DEFine PROCedure EqualsPressed
630 TestShiftPending
640 PRINT ' = ';
650 WorkItOut
660 CALL writenum, 1, accumulator
670 PRINT
680 CALL copynum, CurrentNum, accumulator
690 LastOp = equals
700 END DEFine
710 :
720 DEFine PROCedure OperatorPressed
730 TestShiftPending
740 WorkItOut
750 CALL copynum, DisplayedNum, accumulator
760 CALL SetTo0, CurrentNum
770 LastOp = key
780 PRINT ' '; CHR$(key); ' ';
790 END DEFine
800 :
810 DEFine PROCedure WorkItOut
820 SELect ON LastOp
830   = plus   : call add_, DisplayedNum, CurrentNum
840   = minus  : call subtract, DisplayedNum, CurrentNum
850   = times  : call multiply, DisplayedNum, CurrentNum
860   = divide : call division, DisplayedNum, CurrentNum
870 END SELect
880 END DEFine
890 :
900 DEFine PROCedure TestShiftPending
910 IF ShiftPending = INTEGER THEN
920   CALL ShiftLeft, CurrentNum, Decimals
930 ELSE
940   CALL ShiftLeft, CurrentNum, ShiftPending
950 END IF
960 ShiftPending = INTEGER
970 END DEFine
980 :
```

COS(X) - 1 + 0 - X^2/2! + 0 + X^4/4!

Note the alternating + and - signs between successive terms. We can define TAN(X) = SIN(X)/COS(X), which saves finding a power series expansion for it.

Four functions

Listing four is a four-function calculator. One point to note is the way the program handles the input of decimal numbers. As you type in each digit of a number, it shifts the number left by one place and stores the new digit at the right-hand end. If you come to the end of a number and press the equals key (or whatever) without keying in a decimal point, then the number is an integer and must be shifted left DECIMALS places. If you have keyed in a decimal point, the program must keep a count of the number of digits since the point, and shift the number left by DECIMALS minus this many places.

Sample output of listing 4

```
FOUR FUNCTION CALCULATOR
Maximum integer part  = 60 digits
Maximum fraction part = 20 digits

2 * 2 = 4.00000000000000000000
 * 4 = 16.00000000000000000000
 * 16 = 256.00000000000000000000
 * 256 = 65536.00000000000000000000
 * 65536 = 4294967296.00000000000000000000
 / 1234567.39845769856934563465 = 3478.92492654961640114310

1111111111 * 111111111111111111 = 123456790111111111987654321.00000000000000
11223344 * 44332211 = 497555654333584.00000000000000000000
122333444 * 4444333221 = 543690589208543124.00000000000000000000
555555555 * 444444444 = 246913579753086420.00000000000000000000

3.14159265358979323846 / 2 = 1.57079632679489661923
 / 2 = 0.78539816339744830961
 * 8 = 6.28318530717958647688
 / 2 = 3.14159265358979323844
999999999 * 999999999 = 999999998000000001.00000000000000000000
```

Systematic Machine Code Programmi

In part four of his machine code tutorial, Alan Bridewell adds more screen control commands.

In this series, we are building up a library of small chunks of assembler language code. Each of these will do a recognisable job within a program and be fully annotated, so that it will be clear exactly how to join them to other chunks to make a program and, in particular, how to make any necessary changes to make the chunks fit together.

In this part, we are going to look at some more screen control commands. These are the ones concerned with interesting things like plotting points, lines, arcs, ellipses and circles on the screen. Compared with the other commands, dealt with in part 3, these have two additional complications.

Each screen window has a pair of 'graphics coordinates', x and y. By default, when a window is first opened, the bottom left hand corner of the window is the origin (x=0, y=0). The vertical scale of the window is 100 units, which means the top left hand corner is the point x=0, y=100. The horizontal scale is the same size as the vertical scale, so the maximum x value will depend on the width compared with the height. This is all true, regardless of the size and shape of the window, but these actual values can be altered using the SCALE command. So the origin can be placed at any point in the window (or outside, if required), and the vertical scale can be made any value required.

When we draw points, lines, arcs, ellipses or circles in a window, the parameters we use refer to the coordinate system for the window, as defined by "SCALE', or using the default values if "SCALE' has not been used. For example, this means that when we draw a circle in a window, the size

Listing 1

```
; ***************************************************************************
;                                   'POINT'
; ***************************************************************************
; THIS ROUTINE PLOTS A POINT IN THE WINDOW USING GRAPHICS COORDINATES.
; EACH COORDINATE IS ENTERED AS TWO INTEGERS, AND THE COORDINATE IS THE
; RESULT OF THE F.P. DIVISION OF THE FIRST INTEGER BY THE SECOND.
; THE FOUR PARAMETERS ARE ENTERED IN THE INTEGER TABLE
;
; PUT RI STACK 50 BELOW TOP OF DATA SPACE.
.POINT              LEA.L     -50(A5),A1          ; RI STACK ON A1
;
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1       ; INTEGER = 2 BYTES
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    PXA,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0       ; #RI_FLOAT ON D0
                    MOVE.W    $11C,A3     ; RI_EXEC ON A3
                    JSR       (A3)
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    PXB,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE X COORD ON RI STACK.
                    MOVEQ     #$10,D0     ; #RI_DIV ON D0
                    JSR       (A3)
; USE NEXT LINE IF A NEGATIVE X COORDINATE IS REQUIRED
;                    JSR       NEGATE
;
; NOW REPEAT THE WHOLE THING FOR THE Y COORDINATE
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    PYA,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0       ; #RI_FLOAT ON D0
                    JSR       (A3)
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    PYB,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE Y COORD ON RI STACK.
                    MOVEQ     #$10,D0     ; #RI_DIV ON D0
                    JSR       (A3)
; USE NEXT LINE IF A NEGATIVE X COORDINATE IS REQUIRED
;                    JSR       NEGATE
;
; NOW PLOT THE POINT
                    MOVE.L    (A7),A0     ; OR 4(A7),A0 OR 8(A7),A0 ETC.
                                          ; CONSOLE CHANNEL ID IN A0
                    MOVEQ     #$30,D0     ; #SD.POINT ON D0
                    MOVE.W    #$FFFF,D3   ; INFINITE TIMEOUT
                    ADDA.L    A6,A1       ; MAKE A1 STACK ABSOLUTE
                    TRAP      #3
                    BRA.S     NEXTBIT     ; SKIP OVER INTEGER TABLE, ETC.
;
; INTEGER TABLE
.PXA                DC.W      10          ; 1ST INTEGER OF X
.PXB                DC.W      1           ; 2ND INTEGER OF X
.PYA                DC.W      20          ; 1ST INTEGER OF Y
.PYB                DC.W      1           ; 2ND INTEGER OF Y
;
; SUBROUTINE TO NEGATE F.P.NUMBER TOS ( DELETE IF NOT NEEDED )
.NEGATE             MOVEQ     #$14,D0     ; RI_NEG IN D0
                    JSR       (A3)
                    RTS
;
; ***************************************************************************
```

c
ing

of the circle will depend on the size of the window and its scale, as well as the circle parameters. Using the same circle parameters in different windows will, in general, result in different sized circles.

The second complication is that all these parameters have to be entered as real or floating point numbers, rather than the integers we have used up to now. Because the microprocessor in the QL can only handle integers directly, these commands have to use the Qdos arithmetic routines, which handle floating point numbers.

Why the need for these extra complications? Well firstly, the pixels on the QL screen are not square, and are in fact a different shape in modes 4 and 8. This means that, working in pixel coordinates, it would be rather difficult to draw, say, a perfect circle. If the circle was perfect in one mode, it would no longer be so in the other mode. Using graphic coordinates gets over this problem. Secondly, we often need parameters which are not whole numbers. For instance, to draw a semicircle with the ARC command requires an angle of pi radians, 3.14#. . .

Clearly, we need to use floating point numbers.

Floating points

The graphics traps use a parameter table of floating point numbers, much the same way as, say, the BLOCKFILL command uses a parameter table of word integers. So in theory we should be able to use the same format as before, except we enter floating point parameters in a floating point parameter table. If only things were so simple! The problem is that the floating point representation of numbers that the QL uses bears no obvious relationship to the real number. Entering floating point numbers in a parameter table would be a real pain. So how do we overcome the problem?

My solution is this. Each required number is treated as the result of the division of one integer by another. For instance, pi can be represented as 3142 and 1000. By dividing the first by the second we get 3.142, pi

Listing 2

```
; ************************************************************************
;                                  'LINE'
; ************************************************************************
; THIS ROUTINE PLOTS A LINE IN THE WINDOW USING GRAPHICS COORDINATES.
; EACH COORDINATE IS ENTERED AS TWO INTEGERS, AND THE COORDINATE IS THE
; RESULT OF THE F.P. DIVISION OF THE FIRST INTEGER BY THE SECOND.
; THE PARAMETERS ARE ENTERED IN THE INTEGER TABLE.
;
; PUT RI STACK 50 BELOW TOP OF DATA SPACE.
.LINE               LEA.L     -50(A5),A1          ; RI STACK ON A1
;
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1      ; INTEGER = 2 BYTES
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    LX1A,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0      ; #RI_FLOAT ON D0
                    MOVE.W    $11C,A3    ; RI_EXEC ON A3
                    JSR       (A3)
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    LX1B,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE X COORD ON RI STACK.
                    MOVEQ     #$10,D0    ; #RI_DIV ON D0
                    JSR       (A3)
; USE NEXT LINE IF A NEGATIVE X COORDINATE IS REQUIRED
;                    JSR       NEGATE
;
; NOW REPEAT THE WHOLE THING FOR THE Y COORDINATE
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    LY1A,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0      ; #RI_FLOAT ON D0
                    JSR       (A3)
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    LY1B,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE Y COORD ON RI STACK.
                    MOVEQ     #$10,D0    ; #RI_DIV ON D0
                    JSR       (A3)
; USE NEXT LINE IF A NEGATIVE Y COORDINATE IS REQUIRED
;                    JSR       NEGATE
;
; NOW REPEAT THE WHOLE PROCESS FOR THE COORDS OF THE SECOND POINT
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    LX2A,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0      ; #RI_FLOAT ON D0
                    JSR       (A3)
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    LX2B,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE X COORD ON RI STACK.
                    MOVEQ     #$10,D0    ; #RI_DIV ON D0
                    JSR       (A3)
; USE NEXT LINE IF A NEGATIVE X COORDINATE IS REQUIRED
;                    JSR       NEGATE
;
; NOW REPEAT THE WHOLE THING FOR THE Y COORDINATE
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    LY2A,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0      ; #RI_FLOAT ON D0
                    JSR       (A3)
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    LY2B,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE Y COORD ON RI STACK.
                    MOVEQ     #$10,D0    ; #RI_DIV ON D0
                    JSR       (A3)
; USE NEXT LINE IF A NEGATIVE Y COORDINATE IS REQUIRED
;                    JSR       NEGATE
;
; NOW PLOT THE LINE
                    MOVE.L    (A7),A0    ; OR 4(A7),A0 OR 8(A7),A0 ETC.
                                         ; CONSOLE CHANNEL ID IN A0
                    MOVEQ     #$31,D0    ; #SD.LINE ON D0
                    MOVE.W    #$FFFF,D3  ; INFINITE TIMEOUT
                    ADDA.L    A6,A1      ; MAKE A1 STACK ABSOLUTE
                    TRAP      #3
                    BRA.S     NEXTBIT    ; SKIP OVER INTEGER TABLE, ETC.
```

correct to three decimal places. To get pi on my floating point table, I do the following:

1. Enter 3142 and 1000 as two word integers in an integer table.
2. Use the QL arithmetic package to convert them into floating point numbers.
3. Use the QL arithmetic package to divide one by the other to give 3.142.
4. Arrange that when this has been done, the resulting floating point number is in the correct place in the required floating point parameter table.

Negative parameters

One further complication: we may need our floating point parameter to be negative. The obvious way to deal with this would be to make the first integer negative. Making the first integer -3142 and the second 1000 would give us minus pi. Unfortunately, it doesn't work. To be precise, it seems to work when the required parameter is a negative angle, but not if it is a negative x or y coordinate. Why this should be so is a complete mystery to me. When I assemble the code and trace it in single steps, it always works; if I SEXEC the code and then EXEC the program, it always crashes. Perhaps I have missed something. If you can see what is wrong perhaps you can let me know. On the other hand, perhaps I've found a Qdos bug. I wonder . . .

Anyway, there is another solution which does seem to work. All we do is use positive integers, and then use the arithmetic package to negate the floating point number.

The QL arithmetic package does a great deal which we do not really want to consider at this stage. This will be the subject of a later part when we look at dealing with all the various arithmetical processes it can do. For the moment we shall limit ourselves to what is necessary to handle the graphics traps.

Execution of an arithmetic operation is a 'vectored utility'. The vector $11C is put into an address register, say A3. This vector contains the address of the subroutine, so a JSR (A3) will jump to the routine. For the routine to work properly, other registers must have the correct setting beforehand.

Execution

Register D0 requires a word to say which arithmetic operation is to be executed. For instance, #$10 in D0 will cause a divide. The floating point number on top of the stack will divide into the next floating point number on stack. The floating point result will be left on the stack in place of the two original numbers. However, this is not the normal stack but a special one. More about this in a moment.

Register A4 should point to the base of the data area. We don't need to do anything about this because when we EXEC a program, the base address of the data area automatically goes into A4.

Register A1 needs to point to the special stack needed for the arithmetic data, usually called the RI stack. We have to decide

```
;
; INTEGER TABLE
.LX1A                   DC.W      10          ; 1ST INTEGER OF X1
.LX1B                   DC.W      1           ; 2ND INTEGER OF X1
.LY1A                   DC.W      20          ; 1ST INTEGER OF Y1
.LY1B                   DC.W      1           ; 2ND INTEGER OF Y1
.LX2A                   DC.W      100         ; 1ST INTEGER OF X2
.LX2B                   DC.W      1           ; 2ND INTEGER OF X2
.LY2A                   DC.W      50          ; 1ST INTEGER OF Y2
.LY2B                   DC.W      1           ; 2ND INTEGER OF Y2
;
; SUBROUTINE TO NEGATE THE F.P. NUMBER TOS
;
.NEGATE                 MOVEQ     #$14,D0     ; RI_NEG IN D0
                        JSR       (A3)
                        RTS
;
; ***************************************************************************
```

Listing 3

```
; ***************************************************************************
;                                   'ARC'
; ***************************************************************************
; THIS ROUTINE PLOTS AN ARC IN THE WINDOW USING GRAPHICS COORDINATES.
; THE SD.ARC TRAP REQUIRES 5 PARAMETERS, THE X AND Y COORDINATES OF THE
; ENDS OF THE ARC, AND THE ANGLE (IN RADIANS) SUBTENDED BY THE ARC.
; EACH PARAMETER IS ENTERED AS TWO INTEGERS, AND THE RESULT OF THE F.P.
; DIVISION OF THE FIRST INTEGER BY THE SECOND GIVES THE REQUIRED
; PARAMETER (COORDINATE OR ANGLE).
; THE 10 INTEGERS ARE ENTERED IN THE INTEGER TABLE
;
; PUT RI STACK 50 BELOW TOP OF DATA SPACE.
.ARC                    LEA.L     -50(A5),A1          ; RI STACK ON A1
;
; MAKE ROOM FOR 1ST INTEGER
                        SUBQ.L    #2,A1       ; INTEGER = 2 BYTES
; PUT FIRST INTEGER ON RI STACK
                        MOVE.W    AX1A,0(A6,A1.L)
; CONVERT TO F.P.
                        MOVEQ     #8,D0       ; #RI_FLOAT ON D0
                        MOVE.W    $11C,A3     ; RI_EXEC ON A3
                        JSR       (A3)
; REPEAT FOR 2ND INTEGER
                        SUBQ.L    #2,A1
                        MOVE.W    AX1B,0(A6,A1.L)
                        MOVEQ     #8,D0
                        JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE X COORD ON RI STACK.
                        MOVEQ     #$10,D0     ; #RI_DIV ON D0
                        JSR       (A3)
; USE NEXT LINE IF A NEGATIVE X COORDINATE IS REQUIRED
;                        JSR       NEGATE
;
; NOW REPEAT THE WHOLE THING FOR THE Y COORDINATE
; MAKE ROOM FOR 1ST INTEGER
                        SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                        MOVE.W    AY1A,0(A6,A1.L)
; CONVERT TO F.P.
                        MOVEQ     #8,D0       ; #RI_FLOAT ON D0
                        JSR       (A3)
; REPEAT FOR 2ND INTEGER
                        SUBQ.L    #2,A1
                        MOVE.W    AY1B,0(A6,A1.L)
                        MOVEQ     #8,D0
                        JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE Y COORD ON RI STACK.
                        MOVEQ     #$10,D0     ; #RI_DIV ON D0
                        JSR       (A3)
; USE NEXT LINE IF A NEGATIVE Y COORDINATE IS REQUIRED
;                        JSR       NEGATE
;
; NOW REPEAT THE WHOLE PROCESS FOR THE COORDS OF THE SECOND POINT
; MAKE ROOM FOR 1ST INTEGER
                        SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                        MOVE.W    AX2A,0(A6,A1.L)
; CONVERT TO F.P.
                        MOVEQ     #8,D0       ; #RI_FLOAT ON D0
                        JSR       (A3)
; REPEAT FOR 2ND INTEGER
                        SUBQ.L    #2,A1
                        MOVE.W    AX2B,0(A6,A1.L)
                        MOVEQ     #8,D0
                        JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE X COORD ON RI STACK.
                        MOVEQ     #$10,D0     ; #RI_DIV ON D0
                        JSR       (A3)
```

where to put it. Clearly it has to go somewhere in the program's data area. Register A4 points to the bottom of this area automatically, as we have already said. The top of the data area automatically goes in register A5 when we EXEC a program. Our RI stack must go somewhere between the two. In fact the top of the data area (in A5) is also the top of the normal stack for the program, so we need to leave enough space below that for that stack. However, if we go too far down towards the address held in A4, we may interfere with other uses of the data area. It appears that 50 bytes is more than enough space for the normal stack, so if we put the top of the RI stack at 50 bytes below the address in A5, that should leave enough normal stack space.

The only problem then is to make sure the data space left is big enough for the RI stack plus any other call that may be made on data space. The graphics calls we shall make apparently need at least 240 bytes of RI stack, which makes 50 + 240 = 290 bytes for the two stacks. Leave about the same again for other use of data space and that means we need about 500 bytes at least for data space.

The question of how much data space a program needs is very difficult to analyse without a very detailed knowledge of exactly what Qdos is doing. The best policy is 'suck it and see'. In other words, start off with a big data space. (Assembler Workbench, which I use, defaults to a data space of 4K for SEXEC commands, and that is bigger than I have ever needed.) If everything works OK and if you feel you may need the unused space, you can experiment with smaller data spaces until you find the smallest that seems to work.

The only other point we need to make is about register D7. Apparently this should be set to 0 in Qdos versions 1.03 and earlier, so if you have a very old QL you may need to bear it in mind. However, I shall ignore it.

```
; USE NEXT LINE IF A NEGATIVE X COORDINATE IS REQUIRED
;                    JSR        NEGATE
;
; NOW REPEAT THE WHOLE THING FOR THE Y COORDINATE
; MAKE ROOM FOR 1ST INTEGER
                     SUBQ.L     #2,A1
; PUT FIRST INTEGER ON RI STACK
                     MOVE.W     AY2A,0(A6,A1.L)
; CONVERT TO F.P.
                     MOVEQ      #8,D0      ; #RI_FLOAT ON D0
                     JSR        (A3)
; REPEAT FOR 2ND INTEGER
                     SUBQ.L     #2,A1
                     MOVE.W     AY2B,0(A6,A1.L)
                     MOVEQ      #8,D0
                     JSR        (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE Y COORD ON RI STACK.
                     MOVEQ      #$10,D0    ; #RI_DIV ON D0
                     JSR        (A3)
; USE NEXT LINE IF A NEGATIVE Y COORDINATE IS REQUIRED
;                    JSR        NEGATE
;
; NOW REPEAT THE WHOLE THING FOR THE ANGLE
; MAKE ROOM FOR 1ST INTEGER
                     SUBQ.L     #2,A1
; PUT FIRST INTEGER ON RI STACK
                     MOVE.W     ANGLEA,0(A6,A1.L)
; CONVERT TO F.P.
                     MOVEQ      #8,D0      ; #RI_FLOAT ON D0
                     JSR        (A3)
; REPEAT FOR 2ND INTEGER
                     SUBQ.L     #2,A1
                     MOVE.W     ANGLEB,0(A6,A1.L)
                     MOVEQ      #8,D0
                     JSR        (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE ANGLE ON RI STACK.
                     MOVEQ      #$10,D0    ; #RI_DIV ON D0
                     JSR        (A3)
; USE NEXT LINE IF A NEGATIVE ANGLE IS REQUIRED
;                    JSR        NEGATE
;
; NOW PLOT THE ARC
                     MOVE.L     (A7),A0    ; OR 4(A7),A0 OR 8(A7),A0 ETC.
                                           ; CONSOLE CHANNEE ID IN A0
                     MOVEQ      #$32,D0    ; #SD.ARC ON D0
                     MOVE.W     #$FFFF,D3  ; INFINITE TIMEOUT
                     ADDA.L     A6,A1      ; MAKE A1 STACK ABSOLUTE
                     TRAP       #3
                     BRA.S      NEXTBIT    ; SKIP OVER INTEGER TABLE, ETC.
;
; INTEGER TABLE
.AX1A                DC.W       10         ; 1ST INTEGER OF X1
.AX1B                DC.W       1          ; 2ND INTEGER OF X1
.AY1A                DC.W       20         ; 1ST INTEGER OF Y1
.AY1B                DC.W       1          ; 2ND INTEGER OF Y1
.AX2A                DC.W       100        ; 1ST INTEGER OF X2
.AX2B                DC.W       1          ; 2ND INTEGER OF X2
.AY2A                DC.W       50         ; 1ST INTEGER OF Y2
.AY2B                DC.W       1          ; 2ND INTEGER OF Y2
.ANGLEA              DC.W       3142       ; 1ST INTEGER OF ANGLE
.ANGLEB              DC.W       1000       ; 2ND INTEGER OF ANGLE
;
; SUBROUTINE TO NEGATE THE F.P. NUMBER TOS
;
.NEGATE              MOVEQ      #$14,D0    ; RI_NEG IN D0
                     JSR        (A3)
                     RTS
;
; ************************************************************************
```

Reverse order

This is all (and I think it's quite enough for now!) we need to concern ourselves with about arithmetic operations. We now need to consider the matter of using the arithmetic package to turn our initial integer table into the floating point table that the graphics traps require. This is done by obeying a simple rule. We carry out our arithmetic operations to put items on the stack in the reverse order. This means the last parameter we add, which is the first one we need, will be at the top of the stack and also, therefore, at the top of the parameter.

There is one last complication which needs to be dealt with before we can look at the listings. I said earlier that register A4 points to the base of the data space, and register A5 to the top of the data space. This is not strictly true. What actually happens is this: register A6 points to the base of the job (ie the program area). Registers A4 and A5 contain offsets from this to point to the bottom and top of the data space. So to get the true addresses of the bottom and top of the data space, we need to add the address in A6 to A4 and A5. Since our RI stack was obtained by subtracting a number from register A5 and putting it in register A1, then clearly we need to add the address in A6 to get the true address of our RI stack.

Adding addresses

Up to this point, this fact has been of no real consequence. But now we are going to put data onto the RI stack, ready to be used by the arithmetic package, so we need a command which takes the offset into account. Instead of using the command:

MOVE.W DATA,A1

which we might have expected, we must use:

MOVE.W DATA,0 (A6,A1.L)

which will add the address in A6 to the offset in A1 to find the correct address. Furthermore, when we make our graphics trap call, the trap needs the true address in A1, not the offset. So before making the

trap call, we must add the address in A6 to it, with the command:

ADDA.L A6,A1

There is an alternative to this, by the way. It is to make a TRAP 4 call just before the TRAP 3. TRAP 4 has no parameters and no error returns. I don't use it because it does something slightly different when preceding a TRAP 2 call and a TRAP 3 call and I like, as far as possible, the code to say exactly what it is doing.

Look at listings

Now to look at the listings. As with all the previous screen handling traps, A0 must hold the channel ID, and D3 must hold the timeout. As this is the same for all the listings, we shall not mention it again. We shall look at the first listing in detail, and skip over the same points as they are repeated in the other listings.

POINT **(Listing one)** will plot a point in the window using graphics coordinates. The SD_POINT trap requires the Y coordinate (floating point) on the top of the RI stack, and the x coordinate (floating point) next. We are going to calculate these from two word integers in an integer table, so the table requires two word integers for the x coordinates and two for the y coordinates of the point, making four integers in all.

First we load the address offset for our RI stack in A1.

Next we deal with the x coordinate. (Remember, we do them in reverse order.)

We want to add the first integer to the RI stack. When the arithmetic package uses the RI stack it automatically adjusts the stacker pointer as numbers are added or removed. But if we are going to simply use a command to put data in an address, we must move the RI stack pointer ourselves. So first we subtract 2 from the RI stack pointer to make room for the integer, before moving the first integer onto the RI stack. Then we call the arithmetic package to convert it to floating point. We now repeat this for the second integer (ie move the RI stack pointer, move the next integer onto the RI stack, and convert to floating point).

Negative line

The RI stack now holds an FP version of the second integer on top and an FP version of the first integer below. So we use the arithmetic package to divide one by the other, to leave an FP version of the x coordinate on the RI stack. If we need a negative x coordinate, we use the next line to jump to a subroutine to get the arithmetic package to make it negative. (This line appears in all the listings wherever a parameter might need to be negative – which is most of them! So we shall not mention this again.)

If we repeat this whole process for they

Listing 4

```
; ************************************************************************
;                                 'ELLIPSE'
; ************************************************************************
; THIS ROUTINE PLOTS AN ELLIPSE IN THE WINDOW USING GRAPHICS COORDINATES.
; THE SD.ELIPS TRAP REQUIRES 5 F.P. PARAMETERS, THE X AND Y COORDINATES
; OF THE CENTRE, THE ECCENTRICITY, THE RADIUS, AND THE ROTATION ANGLE.
; EACH PARAMETER IS ENTERED AS TWO INTEGERS, AND THE PARAMETER IS THE
; RESULT OF THE F.P. DIVISION OF THE FIRST INTEGER BY THE SECOND.
; THE 10 INTEGERS ARE ENTERED IN THE INTEGER TABLE.
;
; PUT RI STACK 50 BELOW TOP OF DATA SPACE.
.ELLIPSE            LEA.L     -50(A5),A1          ; RI STACK ON A1
;
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    EXA,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0     ; #RI_FLOAT ON D0
                    MOVE.W    $11C,A3   ; RI_EXEC ON A3
                    JSR       (A3)
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    EXB,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE X COORD ON RI STACK.
                    MOVEQ     #$10,D0   ; #RI_DIV ON D0
                    JSR       (A3)
; USE NEXT LINE IF A NEGATIVE X COORDINATE IS REQUIRED
;                    JSR       NEGATE
;
; NOW REPEAT THE WHOLE THING FOR THE Y COORDINATE
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    EYA,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0     ; #RI_FLOAT ON D0
                    JSR       (A3)
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    EYB,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE Y COORD ON RI STACK.
                    MOVEQ     #$10,D0   ; #RI_DIV ON D0
                    JSR       (A3)
; USE NEXT LINE IF A NEGATIVE Y COORDINATE IS REQUIRED
;                    JSR       NEGATE
;
; NOW REPEAT THE WHOLE THING FOR THE ECCENTRICITY
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    ECCA,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0     ; #RI_FLOAT ON D0
                    JSR       (A3)
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    ECCB,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE ECCEN. ON RI STACK.
                    MOVEQ     #$10,D0   ; #RI_DIV ON D0
                    JSR       (A3)
;
; NOW REPEAT THE WHOLE THING FOR THE RADIUS
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    RADIUSA,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0     ; #RI_FLOAT ON D0
                    JSR       (A3)
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    RADIUSB,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE RADIUS ON RI STACK.
                    MOVEQ     #$10,D0   ; #RI_DIV ON D0
                    JSR       (A3)
;
; NOW REPEAT THE WHOLE THING FOR THE ANGLE OF ROTATION
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    ROTA,0(A6,A1.L)

; CONVERT TO F.P.
                    MOVEQ     #8,D0     ; #RI_FLOAT ON D0
                    JSR       (A3)
```

```
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    ROTB,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE ROTATION ON RI STACK.
                    MOVEQ     #$10,D0    ; #RI_DIV ON D0
                    JSR       (A3)
; USE NEXT LINE IF A NEGATIVE ANGLE IS REQUIRED
;                    JSR       NEGATE
;
; NOW PLOT THE ELLIPSE
                    MOVE.L    (A7),A0    ; OR 4(A7),A0 OR 8(A7),A0 ETC.
                                         ; CONSOLE CHANNEL ID IN A0
                    MOVEQ     #$33,D0    ; #SD.ELIPS ON D0
                    MOVE.W    #$FFFF,D3  ; INFINITE TIMEOUT
                    ADDA.L    A6,A1      ; MAKE A1 STACK ABSOLUTE
                    TRAP      #3
                    BRA.S     NEXTBIT    ; SKIP OVER INTEGER TABLE, ETC.
;
; INTEGER TABLE
.EXA                DC.W      10         ; 1ST INTEGER OF CENTRE X
.EXB                DC.W      1          ; 2ND INTEGER OF CENTRE X
.EYA                DC.W      20         ; 1ST INTEGER OF CENTRE Y
.EYB                DC.W      1          ; 2ND INTEGER OF CENTRE Y
.ECCA               DC.W      2          ; 1ST INTEGER OF ECCENTRICITY
.ECCB               DC.W      1          ; 2ND INTEGER OF ECCENTRICITY
.RADIUSA            DC.W      20         ; 1ST INTEGER OF RADIUS
.RADIUSB            DC.W      1          ; 2ND INTEGER OF RADIUS
.ROTA               DC.W      3142       ; 1ST INTEGER OF ROTATION
.ROTB               DC.W      4000       ; 2ND INTEGER OF ROTATION
;
; SUBROUTINE TO NEGATE THE F.P. NUMBER TOS

;
.NEGATE             MOVEQ     #$14,D0    ; RI_NEG IN D0
                    JSR       (A3)
                    RTS
;
; ***************************************************************************
```

coordinate, we shall finish up with the FP version of the y coordinate on top of the RI stack, and the FP version of the x coordinate below it, which is precisely what is required for the SD_POINT trap.

True stack pointer

So to the final steps. We move $30 into D0 for the SD_POINT trap call. We add the address in A6 to A1 to get the true RI stack pointer in A1, and then make the TRAP 3 call. All being well, the point will now be plotted. Finally, we branch to the next bit of code to skip over the integer table and NEGATE subroutine (if it hasn't been deleted because it was not needed.

LINE **(Listing two)** will plot a line in the window using the graphics coordinates. SD_LINE requires four floating point parameters on the RI stack, as follows, starting from the top of the stack:

y coordinate of the end of the line
x coordinate of the end of the line
y coordinate of the start of the line
x coordinate of the start of the line

As these FP numbers are going to be calculated from two integers each, we need an integer table with eight-word integers on it.

First we load the address offset for our RI stack into A1.

Next we process the integers into FP numbers, in exactly the same way as in POINT. Again, we do them in reverse order, to get the four FP parameters on the RI stack.

Finally we put $31 in D0, correct the RI stack pointer in A1 and make the trap call to plot the line, before branching to the next bit.

ARC **(Listing three)** plots an arc in the window using graphics coordinates. SD_ARC requires five FP parameters on the RI stack, as follows, starting with the top of the RI stack.

angle (in radians) subtended by the arc
y coordinate of the end of the arc
x coordinate of the end of the arc
y coordinate of the start of the arc
x coordinate of the start of the arc

With two integers for each FP number, the integer table contains 10 integers.

As with POINT and LINE, they are processed in reverse order to get the five FP parameters on the RI stack in the correct order.

Trap call

Finally we make the trap call as before, this time with $32 in D0 for SD_ARC trap call.

ELLIPSE **(Listing four)** plots an ellipse or circle in the window using graphics coordinates. SD_ELIPS requires five FP parameters on the RI stack, as follows, starting with the top of the RI stack:

rotation angle (in radians) of the major axis (this is ignored if a circle is plotted)
radius of the ellipse (length of the minor axis)
eccentricity of the ellipse (ratio major/minor axes. (This is 1 for a circle.))
y coordinate of the centre
x coordinate of the centre

Listing 5

```
; ***************************************************************************
;                                 'SCALE'
; ***************************************************************************
; THIS ROUTINE SETS THE SCALE AND ORIGIN FOR GRAPHICS COORDINATES
; THE SD.SCALE TRAP REQUIRES 3 F.P. PARAMETERS, THE X AND Y COORDINATES
; OF THE BOTTOM LEFT CORNER OF THE WINDOW AND THE HEIGHT OF THE WINDOW
; I.E. THE LENGTH OF THE Y AXIS.
; EACH PARAMETER IS ENTERED AS TWO INTEGERS, AND THE PARAMETER IS THE
; RESULT OF THE F.P. DIVISION OF THE FIRST INTEGER BY THE SECOND.
; THE 6 INTEGERS ARE ENTERED IN THE INTEGER TABLE
;
; PUT RI STACK 50 BELOW TOP OF DATA SPACE.
.SCALE              LEA.L     -50(A5),A1          ; RI STACK ON A1
;
; FIX THE HEIGHT OF THE WINDOW
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1      ; INTEGER = 2 BYTES
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    HEIGHTA,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0      ; #RI_FLOAT ON D0
                    MOVE.W    $11C,A3    ; RI_EXEC ON A3
                    JSR       (A3)
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    HEIGHTB,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE HEIGHT ON RI STACK.
                    MOVEQ     #$10,D0    ; #RI_DIV ON D0
                    JSR       (A3)
;
; NEXT FIX THE ORIGIN
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    SXA,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0      ; #RI_FLOAT ON D0
                    JSR       (A3)
```

▶

```
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    SXB,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE X COORD ON RI STACK.
                    MOVEQ     #$10,D0    ; #RI_DIV ON D0
                    JSR       (A3)
; USE NEXT LINE IF A NEGATIVE X COORDINATE IS REQUIRED
;                    JSR NEGATE
;
; NOW REPEAT THE WHOLE THING FOR THE Y COORDINATE
; MAKE ROOM FOR 1ST INTEGER
                    SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                    MOVE.W    SYA,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0      ; #RI_FLOAT ON D0
                    JSR       (A3)
; REPEAT FOR 2ND INTEGER
                    SUBQ.L    #2,A1
                    MOVE.W    SYB,0(A6,A1.L)
                    MOVEQ     #8,D0
                    JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE Y COORD ON RI STACK.
                    MOVEQ     #$10,D0    ; #RI_DIV ON D0
                    JSR       (A3)
; USE NEXT LINE IF A NEGATIVE Y COORDINATE IS REQUIRED
;                    JSR       NEGATE
;
; NOW SET THE SCALE AND ORIGIN
                    MOVE.L    (A7),A0    ; OR 4(A7),A0 OR 8(A7),A0 ETC.
                                         ; CONSOLE CHANNEL ID IN A0
                    MOVEQ     #$34,D0    ; #SD.SCALE ON D0
                    MOVE.W    #$FFFF,D3  ; INFINITE TIMEOUT
                    ADDA.L    A6,A1      ; MAKE A1 STACK ABSOLUTE
                    TRAP      #3
                    BRA.S     NEXTBIT    ; SKIP OVER INTEGER TABLE, ETC
;
; INTEGER TABLE
.HEIGHTA            DC.W      500        ; 1ST INTEGER OF HEIGHT
.HEIGHTB            DC.W      1          ; 2ND INTEGER OF HEIGHT
.SXA                DC.W      10         ; 1ST INTEGER OF X
.SXB                DC.W      1          ; 2ND INTEGER OF X
.SYA                DC.W      20         ; 1ST INTEGER OF Y
.SYB                DC.W      1          ; 2ND INTEGER OF Y
;
; SUBROUTINE TO NEGATE THE F.P. NUMBER TOS
;
.NEGATE             MOVEQ     #$14,D0    ; RI_NEG IN D0
                    JSR       (A3)
                    RTS
;
; ********************************************************************
```

With two integers for each FP number, the integer table contains 10 integers.

As with POINT, LINE and ARC, they are processed in reverse order to get the five FP parameters on the RI stack in the correct order.

Finally we make the trap call as before, this time with $33 in D0 for SD_ELIPS trap call.

Coordinate change

SCALE **(Listing five)** will alter the graphic coordinates of the bottom left corner of the window, and also the graphics units for the height of the window. By default, these are set to (0,0) and 100 respectively when the window is first opened. SD_SCALE requires three FP parameters on the RI stack, as follows, starting with the top of the RI stack.

y coordinate of the bottom of the window
x coordinate of the left hand edge of the window
length of the y axis

With two integers for each FP number, the integer table contains six integers.

As with the other listings, they are processed in reverse order to get the three FP parameters on the RI stack in the correct order.

Finally we make the trap call as before, this time with $34 in D0 for SD_SCALE trap call.

Pixels take a fix

GCURSOR **(Listing six)** is slightly different in format from the others. It sets the position of the cursor in the window using a combination of graphics coordinates, and pixel coordinates. It works like this: a pair of graphics coordinates fixes a point in the window. (Remember, the graphics origin may not even be on the screen.) Then, having fixed this point, a pair of pixels offsets from this point will set the position of the cursor. SD_GCUR requires four FP parameters on the RI stack, as follows, starting with the top of the RI stack.

x coordinate to be fixed in the window
y coordinate to be fixed in the window
pixel offset to the right of the fixed point
pixel offset below the fixed point

Now pixel offsets will clearly be integers (you can't move 25.63 pixels!) so although they are in FP format, they can each be obtained by simply converting one integer into FP format.

With two integers for each FP graphics coordinate, and one integer for each pixel offset, the integer table contains six integers.

As with the other listings, they are proc-

Listing 6

```
; ********************************************************************
;                              'GCURSOR'
; ********************************************************************
; THIS ROUTINE SETS THE GRAPHICS CURSOR POSITION.
; THE SD.GCUR TRAP REQUIRES 4 F.P. PARAMETERS, X AND Y COORDINATES OF A
; POINT IN THE WINDOW, AND THE RIGHT AND DOWN PIXEL OFFSETS OF THE CURSOR
; FROM THIS POINT.
; THE PIXEL OFFSETS ARE ENTERED AS ONE INTEGER EACH.
; EACH COORDINATE IS ENTERED AS TWO INTEGERS, AND THE COORDINATE IS THE
; RESULT OF THE F.P. DIVISION OF THE FIRST INTEGER BY THE SECOND.
; THE 6 INTEGERS ARE ENTERED IN THE INTEGER TABLE.
;
; PUT RI STACK 50 BELOW TOP OF DATA SPACE.
.GCURS              LEA.L     -50(A5),A1          ; RI STACK ON A1
;
; FIX PIXEL OFFSET DOWNWARDS
; MAKE ROOM FOR INTEGER
                    SUBQ.L    #2,A1
; PUT INTEGER ON RI STACK
                    MOVE.W    DOWN,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0      ; #RI_FLOAT ON D0
                    MOVE.W    $11C,A3    ; RI_EXEC ON A3
                    JSR       (A3)
; USE NEXT LINE IF A NEGATIVE (UP) OFFSET IS REQUIRED
;                    JSR       NEGATE
;
; NEXT FIX PIXEL OFFSET TO THE RIGHT
; MAKE ROOM FOR INTEGER
                    SUBQ.L    #2,A1
; PUT INTEGER ON RI STACK
                    MOVE.W    RIGHT,0(A6,A1.L)
; CONVERT TO F.P.
                    MOVEQ     #8,D0
                    JSR       (A3)
```

▶

```
; USE NEXT LINE IF A NEGATIVE (LEFT) OFFSET IS REQUIRED
;                    JSR       NEGATE
;
; NEXT FIX THE COORDINATES
; MAKE ROOM FOR 1ST INTEGER
                     SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                     MOVE.W    GYA,0(A6,A1.L)
; CONVERT TO F.P.
                     MOVEQ     #8,D0
                     JSR       (A3)
; REPEAT FOR 2ND INTEGER
                     SUBQ.L    #2,A1
                     MOVE.W    GYB,0(A6,A1.L)
                     MOVEQ     #8,D0
                     JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE X COORD ON RI STACK.
                     MOVEQ     #$10,D0    ; #RI_DIV ON D0
                     JSR       (A3)
; USE NEXT LINE IF A NEGATIVE Y COORDINATE IS REQUIRED
;                    JSR       NEGATE
;
; NOW REPEAT THE WHOLE THING FOR THE X COORDINATE
; MAKE ROOM FOR 1ST INTEGER
                     SUBQ.L    #2,A1
; PUT FIRST INTEGER ON RI STACK
                     MOVE.W    GXA,0(A6,A1.L)
; CONVERT TO F.P.
                     MOVEQ     #8,D0      ; #RI_FLOAT ON D0
                     JSR       (A3)
; REPEAT FOR 2ND INTEGER
                     SUBQ.L    #2,A1
                     MOVE.W    GXB,0(A6,A1.L)
                     MOVEQ     #8,D0
                     JSR       (A3)
; NOW DIVIDE ONE F.P. NUMBER BY THE OTHER TO LEAVE Y COORD ON RI STACK.
                     MOVEQ     #$10,D0    ; #RI_DIV ON D0
                     JSR       (A3)
; USE NEXT LINE IF A NEGATIVE Y COORDINATE IS REQUIRED
;                    JSR       NEGATE
;
; NOW SET THE GRAPHICS CURSOR
                     MOVE.L    (A7),A0    ; OR 4(A7),A0 OR 8(A7),A0 ETC.
                                          ; CONSOLE CHANNEL ID IN A0
                     MOVEQ     #$36,D0    ; #SD.POINT ON D0
                     MOVE.W    #$FFFF,D3  ; INFINITE TIMEOUT
                     ADDA.L    A6,A1      ; MAKE A1 STACK ABSOLUTE
                     TRAP      #3
                     BRA.S     NEXTBIT    ; SKIP OVER INTEGER TABLE, ETC
;
; INTEGER TABLE
.DOWN                DC.W      50         ; PIXEL OFFSET DOWNWARDS
.RIGHT               DC.W      10         ; PIXEL OFFSET TO THE RIGHT
.GXA                 DC.W      10         ; 1ST INTEGER OF X
.GXB                 DC.W      1          ; 2ND INTEGER OF X
.GYA                 DC.W      20         ; 1ST INTEGER OF Y
.GYB                 DC.W      1          ; 2ND INTEGER OF Y
;
; SUBROUTINE TO NEGATE THE F.P. NUMBER TOS
;
.NEGATE              MOVEQ     #$14,D0    ; RI_NEG IN D0
                     JSR       (A3)
                     RTS
;
; ********************************************************************
```

essed in reverse order to get the four FP parameters on the RI stack in the correct order.

Finally, we make the trap call as before, this time with $36 in D0 for SD_GCUR trap call.

Fill command

FLOOD **(Listing seven)** turns the graphics fill command on and off. When on, non re-entrant shapes are filled with the ink colour. Once an enclosed shape has been filled, the command automatically switches off, and needs to be called again to fill the next shape. This is a very simple trap, much like the screen handling traps dealt with in part 3. The reason it has been left until now is because it only works with shapes drawn using graphics coordinates, so it fits in here.

The trap requires $35 in D0 for SD_FLOOD, and a long word in D1 to set the status. This is 1 to turn the flood on, and 0 to turn the flood off.

And there it is. By adding these commands to the programs put together in parts 2 and 4 of this series, you can draw shapes on the screen to your heart's content. However, there are one or two final points that need to be mentioned.

Those of you familiar with Qdos will realise that I have avoided using an alternative arithmetic package utility called RI_EXECB. It works like RI_EXEC, except that it allows a whole list of arithmetic operations to be executed in order. Since this is precisely what we have been doing using RI_EXEC, RI_EXECB should have made our chunks of code more concise and, presumably, faster. There are two reasons why I have not used it. Firstly, these chunks were actually put together from smaller chunks to use the RI.EXEC command, and I find this modular approach more systematic, and easier to understand, even if the result is not optimised for brevity and speed. Secondly, RI.EXECB is undoubtedly a deal more difficult to code (at least, I find it so).

Simple shapes

The last point is this: these chunks of code have been designed to do fairly simple tasks of drawing shapes from data in fixed parameter tables, but there are situations where this will not do. We may, for example, wish to use SD.LINE to plot a graph, where the program calculates each point in turn, and then uses SD.LINE to draw from the previous point to the next. Alternatively, we may wish to draw shapes taking information from the keyboard, either keying in actual numbers as parameters, or even using the cursor keys to fix the points to be used (like the 'rubber banding' in CAD programs). Clearly, as things stand, we will have difficulty using these chunks of code in these ways. Still, we can't run before we walk, and later we shall look at possible ways of dealing with these problems.

Next time, we shall look at making (and stopping) sounds on the QL, and reading keypresses. That is, we shall look at using the IPC 8049 microprocessor built into your QL. Happy coding!

Listing 7

```
; ********************************************************************
;                              'FLOOD'
; ********************************************************************
; THIS ROUTINE TURNS GRAPHICS 'FILL' ON OR OFF DEPENDING ON THE LONG
; WORD IN D1.
; D1.L = 0       GRAPHICS FILL OFF
; D1.L = 1       GRAPHICS FILL ON
; THIS IS AUTOMATICALLY SWITCHED OFF AFTER EACH GRAPHICS FILL.
;
-FLOOD               MOVE.L    (A7),A0    ; OR 4(A7),A0 OR 8(A7),A0 ETC.
                                          ; CONSOLE CHANNEL ID IN A0
                     MOVEQ     #$35,D0    ; #SD.FLOOD ON D0
                     MOVE.L    #$1,D1     ; GRAPHICS FILL ON
                     MOVE.W    #$FFFF,D3  ; INFINITE TIMEOUT
                     TRAP      #3
;
; ********************************************************************
```

MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS

For Sale, 2 QL's (JM), Sandy SuperQboard, twin 31/2" drives, Microvitec Colour Monitor, Tandata QConnect Modem, Juki 6100 daisywheel printer, SC1500L dot matrix printer, extensive software including Solution, Professional Publisher, Lightning, The Editor, Text87, Flashback, Taskmaster, TechniQL, GraphiQL, Professional Astrologer, etc. Many manuals and textbooks. £500 ono. Telephone any time 0753 543713.

MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS

QL ROMS AND SPARES

ROMS, MGE (Spanish, MGF (French), MGG (German), JS (English), £20 each. JM (English), £10

MDVS. Complete QL MDV Subunit £20, head and chassis £10, Microswitch £2, Motor £6. Rubber & Plastic Roller £2. MDV2 Stand-off spacer + srew £2. ULA £5. Tr 4,5,6,7 set £2. MDV pcb inc, ULA £11. Prices include UK postage. Send SAE for price list of all QL spares.
Joe Atkinson, 36 Ranelagh Road, Ealing, London W5 5RJ (Mail Order Only).

MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS

For Sale, QL Minerva 512K Ramplus Qdisc 3 interface Toolkit 2, Rom battery backed clock, Tandata Modem Set collection software and books, assembler, QLWorld 1984-1992, some spares, £200. Phone Chris 0992 442807.

QL REAL TIME DIGITIZER

Digitize moving pictures in real time. Transforms QL into a video printer or 3-D image digitizer via a video camera/recorder
* Digitizes in .02 secs.
* 256x256 resolution in 8 levels
* Full, 1/4 or 4x 1/4 screen modes
* Screen update - 5 or 2.7 frame/sec
* 1:1 grey level 9-pin printer dump.
* Interfaces to QL ROM PORT
* Runs on unexpanded QL and multitasks.
* P&P 3.00 UK, 9.00 overseas, state MDV or FLP

£99

CL SYSTEMS, 403 CHAPTER ROAD, DOLLIS HILL LONDON NW2 5NG TEL: 081 459 1351

MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS
MICRO ADS

If you wish to advertise in QL WORLD contact Jason on 071 490 7161

QTop by **COWO** ELECTRONIC

QTop The definitive User-Front-End V1.11 SFR 70.--
The QL's equivalent to ***WINDOWS*** on MS-DOS PC's. The system-manager offers the mainmenus **DESK**, **DEVICES**, **FILES**, **JOBS**, **KEYBOARD**, **PROGRAMS**, **MY PROGS**, **TOOLS** and **OPTIONS** with over 110 sub-menus. *QTop* is ideal for Gold-Card or Harddisc users as it supports real Sub-Directories. The powerful QJUMP extended Pointer Environment is included and all its facilities are supported. Comprehensive 41 page manual.
ArcED The professional Text-Editor V2.6 SFR 39.--
The Text-Editor especially for programmers. Written in fast, reentrant machine code (only 20KB). Comprehensive 29 page manual.
ExeQtor The Super-QL in a limited edition/ask for quotation.
We use the QL PCB, GOLDCARD, QIMI & Mouse, JFC K90 & MF-2 Keyboard, up to 3 discdrives and a harddisc (option) to build the Super-QL in a Tower case. *QTop* and *ArcED* are included on the system disc. Ask for free product information. Software is available on 3.5" or 5.25" disc. All our products are compatible with all QL compatible computer. Free updates and user-support. Please specify English or German version. Prices including P&P. Send orders with cheques to drawn in SFR to:

COWO Electronic
Urs König, Münsterstrasse 4, CH-6210 Sursee, Tel 045-211478
or contact TK Computerware (UK) and QLympic Computer Systems (Germany)

CLASSIFIED

Please send coupon to: Advertising Department Sinclair/QL World, Panini House, 116/120 Goswell Road, London EC1. Pleae place this Classified Advertisement in the next issue of Sinclair/QL World.

NAME .....

ADDRESS .....

.....

DAYTIME TELEPHONE NO .....

I enclose a cheque/P.O. to the value of £17.65 (inc. VAT) for a lineage advertisement of no more than 30 words. Please make cheques payable to MCPC Ltd

DJC

Dilwyn Jones Computing

41 Bro Emrys, Tal-y-Bont,
Bangor, Gwynedd LL57 3YT
Tel: Bangor (0248) 354023

PAGE DESIGNER 2 PLUS £40.00

WRITTEN BY DILWYN JONES

QL desktop publishing made simple with this long awaited revision of the Page Designer 2 program, previously sold by Sector Software.

Mix standard text, high resolution text, and graphics on documents up to 9 pages long (depending on available memory size). Use it to make posters, leaflets, booklets, newsletters, and so on. Pages can be saved, loaded, printed, copied, etc. and can be in up to 4 colours, if you have a colour printer (driver for Star Colour LC10 included).

* Very easy to use via simple on screen menus
* Import wordprocessor text into columns across pages.
* Text direct from either Quill "_doc" or plain text files
* Cut and Paste for moving things around and making clipart
* Clip Art pictures supplied on separate disks
* Large range of fonts supplied
* Improved high resolution text - larger fonts now possible
* Now Gold Card compatible
* New, improved manual
* Many new and improved features
* Retains popular features of previous versions
* Can still use files from older versions
* Conversion routines for fonts & pages from other programs
* Flexible, reconfigurable printer driver
* Configurable startup - page size, fonts, etc
* Graphics section - lines, patterns, circles, etc
* Load screens from art programs like The Painter

PAGE DESIGNER 2 PLUS requires a minimum of 512k RAM to operate. It is available on floppy disk (3.5" or 5.25") only. PAGE DESIGNER 2 PLUS is much enhanced compared with the original, but still costs only £40.00!

UPGRADES - £20.00. If you have an older version of Page Designer 2 and would like to upgrade it to the "2 PLUS" version, please send us the original master disk or proof of purchase from Sector Software (it will be returned to you) with an upgrade fee of £20.00. We will send you the new version on new disks with a new manual.

PAGE DESIGNER 2 PLUS IS SCHEDULED FOR RELEASE IN MID-JANUARY, BUT YOU MAY PREFER TO CONTACT US BEFORE ORDERING IN CASE OF DEVELOPMENT DELAYS. WE WILL TAKE YOUR DETAILS AND CONTACT YOU WHEN IT IS READY IN THIS CASE. PLEASE CONTACT US IF YOU WOULD LIKE AN INFORMATION SHEET CONTAINING FURTHER INFORMATION AND A SAMPLE PRINTOUT FROM THIS SUPER PROGRAM.

BIBLE TEXT DISKS £20.00

King James Bible text available on a set of disks in either Quill "_doc" file format, or in plain text (editor) format. *Please specify which format you would prefer when ordering.*

LIBERATION SOFTWARE PRODUCTS

QLIBERATOR V3.35 £50.00

This great QL SuperBASIC compiler is now available from DJC. QLiberator is compatible with the full range of SuperBASIC commands and functions, and Minerva compatible (even with 2 screen mode). Produce faster, multitasking code from your BASIC programs. Version 3.35 can produce ROMable, re-entrant code, libraries of compiled procedures and functions, and can even produce toolkit routine to expand the BASIC interpreter! Support for error trapping in your programs and even the WHEN VARiable construct where implemented. Requires minimum of 256k memory.

BUDGET QLIBERATOR £25.00

Budget price, not a budget specification! This is a full function, easy to use SuperBASIC compiler that converts slow QL BASIC programs into fast multitasking jobs without fuss. COMPACT DESIGN MAKES IT SUITABLE FOR USE ON AN UNEXPANDED QL. Full language compatibility and useable with commercial toolkits, support for pipes, error trapping, etc.

QLOAD 7 QREF £15.00

QLOAD is a utility for speeding up the loading and saving of SuperBASIC programs on the QL. Even long BASIC programs load in seconds rather than the minutes it could normally take! Even allows the use of machine code extensions in a program. QREF is a BASIC cross-reference utility, a boon for BASIC programmers. Prints sorted lists of variables, procedures, functions etc in a program. Wild card selection of names, and a FIND function for the BASIC editor (or Toolkit 2's ED). We use QLOAD and QREF and would not be without them when programming in QL SuperBASIC.

IMAGE PROCESSOR V2 £15.00

WRITTEN BY DILWYN JONES

Screen picture image processing program. Features computerised Image Enhancement, Edge Detection, Screen Mode Conversion, Recolouring, Decolouring, Pixel Editing, Brush for hand alterations or simple drawing, very useful for tidying up digitised pictures or other applications where you need to alter and tidy up screen pictures. All facilities in Version 2 now work in both mode 4 and mode 8.

UPGRADE FROM OLDER VERSIONS £10.00 - send us your master disk or proof of purchase (it will be returned to you) with upgrade fee of £10.00 and we will supply you with a new disk and manual.

SEE OUR MAIN ADVERT ELSEWHERE IN THIS ISSUE FOR OUR RANGE OF QL PRODUCTS AND POSTAGE RATES OVERSEAS (DJC SOFTWARE IS SENT POST FREE TO UK ADDRESSES) AND PAYMENT METHODS, INCLUDING CREDIT CARDS ACCEPTED.

PERFECTION
PERFECTION PLUS

Perfection is the finest word processor available for any computer. We have received dozens of letters from happy users saying just this... and all of these letters were unsolicited. "Superb" was used most often.

Perfection manages to achieve all the sophistication of the most complex PC word processors while still using a user interface as friendly as Quill's. Perfection has a dual system of user control: menus while you are familiarising yourself with the program, and direct commands for the time when you feel ready for more adventurous things. The two systems can be used interchangeably and even simultaneously. Even more exciting – both systems are iterative. In case you don't understand what this means, let us give you an example: suppose you wished to move a block of text using the menus. You would choose Block Move (yes, it is right in the first menu) and the screen would then tell you to move your cursor to the start of the block. On most word processors you would have to navigate manually to this position: indeed, on many of them (Quill included) only a subset of the normal navigation commands would be available. On Perfection, not only can you use all the manual navigation commands (viz all 28 permutations of CTRL, ALT, SHIFT and the arrow keys!) but in addition you can use direct commands like GoTo Line or Page or any of eight markers. Even more amazingly, you can use Search (either as a direct command or from the menus) *even though you are already 'within' a menu option.*

Perfection has about 200 commands, but the layout of menus and the choice of keys for the direct commands makes it very easy to master. Though a 100+ page manual is provided (with all the important bits right at the front), you should only need to consult it for specialised operations like macros.

Even if speed is not particularly important to you, we assure you that Perfection's lightning performance will enable you to use the word processor in sensible ways that you would not have dreamed possible before. For example, scrolling 100 pages or so is accomplished so quickly using the normal navigation commands that you do not need to bother using a menu option to do the move. Spellchecking, assuming you have Perfection Plus, is accomplished virtually instantly: to spellcheck this whole ad (all the pages) would take under 1.5 seconds... Searching (you can switch case sensitivity, as well as equivalences between tabs, soft spaces and hard spaces) is at the rate of about 100 A4 pages per second.

Moving from one word processor to another is usually very traumatic. With Perfection, this will not be the case. Not only can Perfection read in Quill _doc and _exp files directly (you do not even need to tell it they are Quill files!) but it can make direct and immediate use of your existing Quill printer driver. File re-export is also possible.

Perfection is truly WYSIWYG: this means that bold appears bold on screen, italics appear as italics, underlined as underlined, and so on. Of course, your printer may have functions we do not know about (upside down?). To deal with these, Perfection provides a number of on-screen shaded strips: these can be attached to any printer function you wish, and will not upset justification as a translate would. Of course, translates are provided as well!

A variety of statistics on the document being processed are available: some of them are on view all the time, the rest can be toggled to instantly. Not only is there a word count, but also page, line, character and special character (like Superscript Off) counts. There are also a dozen status indicators, letting you know whether you are in Insert or Overwrite mode, whether a block is defined, whether interactive spellchecking is enabled etc. Current line (from top as well as within page) and column positions and character codes are also available.

A terrific feature of Perfection is the dual screen mode. You can view one part of the document while editing another. The sizes of the two windows are themselves adjustable, both in real-time or via the configurator. We should devote more space to the configurator: however, it must suffice to say that everything that could be dynamically set within Perfection may also be preset with the configurator. The configurator can, for example, allow you to select any of 256 colours for any of a dozen parameters (like paper colour, border colour, status window ink and paper colour etc).

Perfection is fully multitasking without need for any external accessory: however, if you already use QPAC or Taskmaster or similar and are happy, you may go on doing so.

There is absolutely no way that we can prepare you for the quality 'feel' of Perfection. We have a great deal of experience using PC word processors costing many hundreds of pounds: with absolutely no exception, Perfection is far easier to use and master.

So if you thought Perfection was unattainable, you have a very pleasant surprise coming to you!

LIGHTNING SPECIAL EDITION
LIGHTNING

These programs accelerate QL operation by up to 10x (2x –4x is typical) without having any adverse effect whatsoever on compatibility or anything else. Lightning SE is typically 40% faster than the standard version. This acceleration is totally independent of, and in addition to, any speed-up obtained by hardware means. So if you have Gold Card, your need for Lightning SE is just the same as if you had only an unexpanded QL – Lightning SE will accelerate both by the same ratio.

The Lightning programs achieve their acceleration by automatically paging out sections of the QL's operating system and replacing these with optimal, concise code written by us.

Lightning installation is a completely automatic and one-off: no knowledge of computing or programming is required. Once installed, Lightning can be completely forgotten about – you will soon get used to the superb speed! Knob twiddlers are catered for too.

Lightning technology is not built in to any of our other programs. Perfection users (as well as users of all other QL software) should therefore use Lightning all the time.

In summary: if you do not have Lightning, you are wrong. Buy this one FIRST OF ALL!

PROFESSIONAL PUBLISHER

The Professional in Professional Publisher refers to the quality of output from that program, and is not meant to suggest any complexity of operation. Few programs are as easy to use as this one: > 99% of users will be able to do without using a manual!

DIGITAL PRECISION LTD

Professional Publisher is by far the best DTP program for the QL. It is fully compatible with Perfection, Editor, Quill, Eye-Q & the ASCII editors. It allows you to *both* create and import *both* text and graphics. Text can be 'poured' into boxes of any shape , size and number, automatically maintaining justification and hyphenation settings. So flowing text around graphics is a doddle.

Professional Publisher is supplied with a generous selection of fonts of various sizes, as well as clip

Justification is by pixel, not by character. This gives a much smoother effect.

It is pointless for us to try to list all of Professional Publisher's features – we would end up filling half the magazine! We will concentrate on just a few 'points': Professional Publisher is extremely precise, performing all its computations accurate to a small fraction of a millimetre. All its features can be preset by you using its configurator, ruling out the need for repetitive key strokes.

The program is extraordinarily versatile while remaining intuitive in its user interface. Buy it!

PROFESSIONAL PUBLISHER
TOOLBOXES

Toolbox I is an excellent collection of high definition fonts, clip art and utility programs for Professional Publisher. While the fonts supplied with Professional Publisher are excellent, many users will feel the need for a wider range of typefaces and styles.

Toolbox II starts where Toolbox I leaves off, providing an even better – and different – font collection.

The two Toolboxes complement each other and are available together at a special price.

FONT ENLARGER
GRAFIX

Font Enlarger does exactly what you would expect it to from its name. While Professional Publisher is also capable of enlarging fonts, it does them 'on the fly' and consequently is not able to remove the jaggedness caused by magnification. Font Enlarger is much cleverer, and enhances detail without any step effect.

While the built-in printer driver for Professional Publisher is excellent with 9-pin printers, it is not optimal with 24-pin or laser printers. Grafix is.

DIGITAL PRECISION LTD

EYE-Q
ULTRAPRINT

Eye-Q is the finest graphics program for the QL. While there may be other graphics programs with a few more features, no other program comes anywhere close to Eye-Q in sheer enjoyability. Eye-Q develops a pleasurable tactile relationship with you, and makes you feel like an artist (even if you aren't). Eye-Q graphics can be read in by Professional Publisher, and the latter's pages can be exported to Eye-Q (using Toolbox I). Everything in Eye-Q is menu-driven and there is context-sensitive help.

While Eye-Q has its own printer driver, Ultraprint allows you 22 distinct styles/sizes of printer output. The reasoning is that the scale of gradation suitable for pictures is probably unsuitable for text or line drawings.

PC CONQUEROR
SOLUTION

PC Conqueror makes your QL into a PC-compatible machine,automatically. It does this by software means only, so there are no screws to undo or wires to fiddle with. Your QL stays a QL too.

Why, might you ask, should you wish to make your QL into a PC-compatible? The reason is simple: you may wish to run the same programs at home as you do at work. Alternatively, you may wish to tap into the vast storehouse of PC software of every type and description you could imagine.

Using PC Conqueror could not be easier. Just boot up your machine with the PC Conqueror disk in floppy 1 and within 10 seconds your QL will be transformed into a PC that is just waiting to be switched on. From this point on you will do exactly the same as you would if you were running a 'real' PC – this means putting a DOS disk (any version) into one of your drives and pressing a key. If you do not already have legal access to a copy of DOS, we can provide you with one at reasonable cost (see our price list).

PC Conqueror runs as fast as it is possible for a PC emulator to run: we have used all our skills to make it work quickly. Of course, you can make the emulation must faster by using Gold Card and Lightning SE. With this combination, you should get speed noticeably better than that of a PC XT...

PC Conqueror allows you to fine-tune the operating environment of the PC in order to improve performance. If you get a hard disk or other high capacity floppy system, you can utilise part or all of it as a PC hard disk.

PC Conqueror occupies under 80K and leaves 667K free for DOS when run on a Trump Card. This is more than you will get on a 'real' PC.

Solution does what Conqueror does but is about half as fast and is not quite as compatible.

SPELLCHECKER
MEGA DICTIONARY

Spellchecker is what makes Perfection into Perfection Plus. We have made it available as a separate item for two reasons: (a) to allow Perfection owners to add it later (b) to allow users of other word processors to benefit from the very best in spellchecking technology.

Spellchecker is supplied complete with three dictionaries of differing sizes as well as a system for building, reviewing and maintaining user dictionaries.

Spellchecker's ultimate accessory is the Mega Dictionary, which gives the user a vocabulary of over 350,000 words!

3D PRECISION CAD SYSTEM

This program allows you to manipulate shapes and figures in 2D and 3D at a speed that will leave you breathless. Irrespective of whether your interest is in CAD, in animation or in just having fun, this program should not be missed. You can output to plotters directly from it, or alternatively create graphics screens to be manipulated and output by Eye-Q, Ultraprint or Professional Publisher.

SUPER SPRITE GENERATOR

SSG moves things about the screen very fast and very smoothly, without flicker. Sprites can have up to 16 frames.